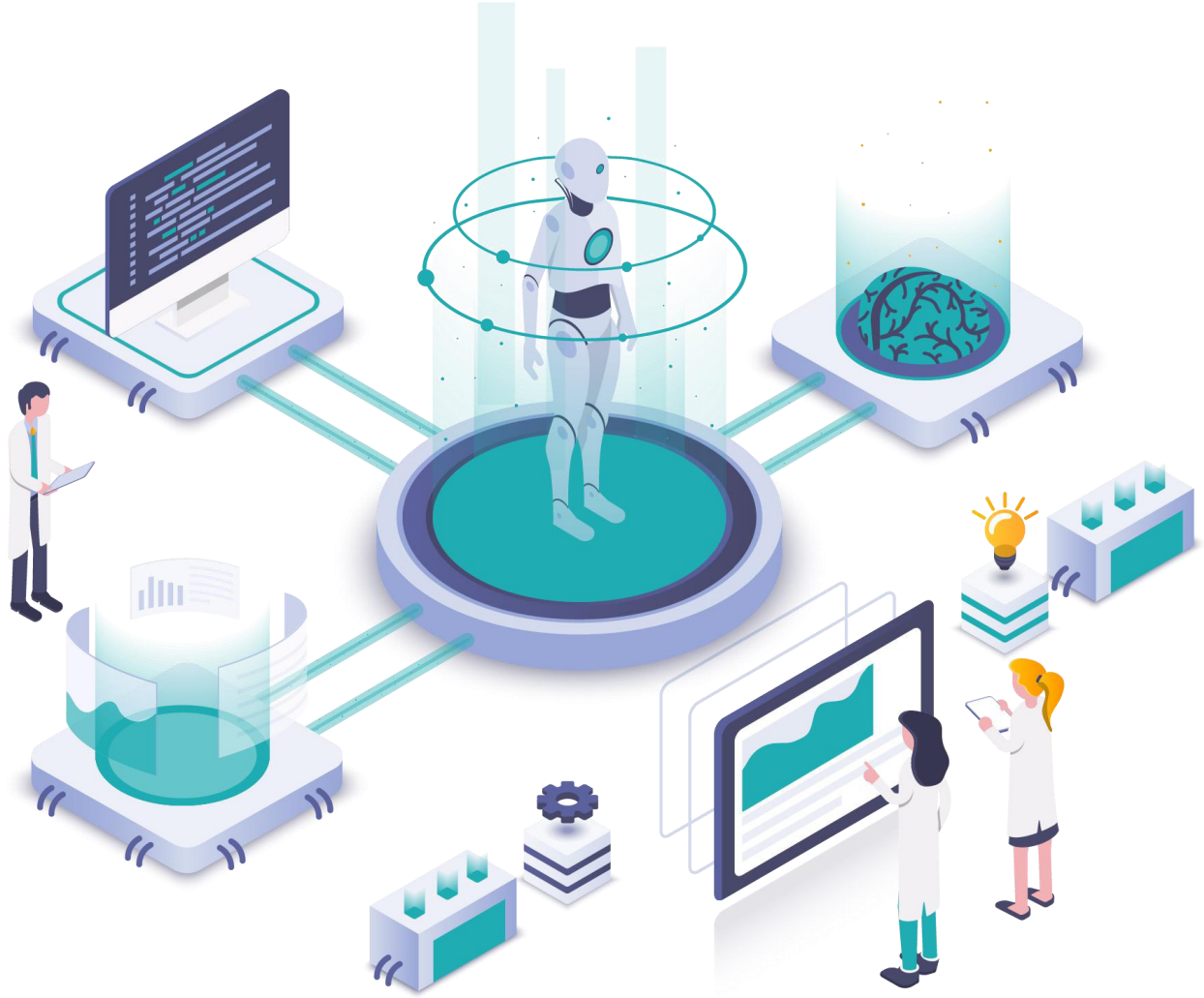
UNIVERSITÉ Clermont Auvergne

académie Clermont-Ferrand

Région académique AUVERGNE-RHÔNE-ALPES



Intelligence Artificiel

DUT - INFO
2A – G5

PERALDE François
SEVRET Yannis

Année : 2020 / 2021

Table des matières

1 – Présentation du sujet

Introduction :

La reconnaissance d'image est de plus en plus utilisée de nos jours, que ce soit par exemple pour déverrouiller son téléphone ou encore pour faire rouler des voitures autonomes. Dans ce travail en plein expansion, nous allons nous intéresser à ce qu'est la reconnaissance d'image et comment cela fonctionne, puis nous verrons de manières plus approfondies comment créer un programme de reconnaissance d'image.

Description du projet :

Le but de ce projet a été de découvrir le langage python, les intelligences artificielles et les réseaux de neurones, un modèle permettant de résoudre des problèmes relevant de l'intelligence artificielle. Suivant nos idées, nous avons décidé de réaliser un programme permettant la reconnaissance de fruits. Cela a été possible par le fruit d'un travail commun entre deux personnes : Sevret Yannis et Peralde François.

Le problème de reconnaissance de fruits peut être qualifié de problème de classification. En effet, le programme doit associer une classe (un nom) à une donnée (une image). L'objectif final de notre projet étant de fournir un programme capable de prendre en entrée une image représentant un fruit et de retourner en sortie le nom de ce fruit.

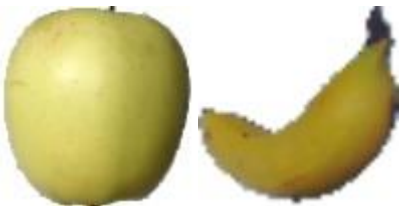
Base de données :

Notre base de données contient des images de 82 fruits différents vu sous différents angles, allant des pommes au raisins passant par les bananes et les litchis. Chaque dossier de fruits contient environ 490 images. Ce qui fait que notre base de données contient environ 40'000 images.

Pour la qualité de nos données, chaque images contenus dans notre base de données ont la même taille, à savoir du 100 par 100. Pour utiliser simplement nos images, nous les transformons en matrices que nous exploitons.

Reconnaissance d'image :

Prenons en premier lieu, un cas très simple, où l'on demande à notre programme de reconnaissance d'image de différencier les images de pomme et de banane.

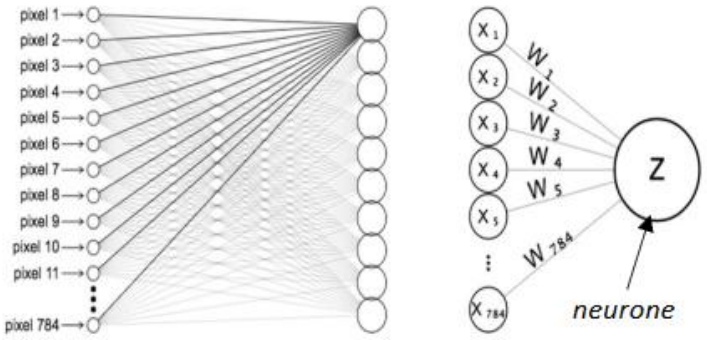


Il est facile pour nous, humains, de différencier ces deux images, respectivement d'une pomme et d'une banane. Cela nous semble évident car nous avons appris à analyser les informations que notre rétine envoie à notre cerveau dès notre plus jeune âge, et nous pouvons même reconnaître des images bien plus complexes comme des visages par exemple. Mais pour apprendre à notre ordinateur à reconnaître des images, il va falloir comprendre de manière plus approfondie ce qu'est réellement la reconnaissance d'image. Reprenons une image de pomme et une image de banane :

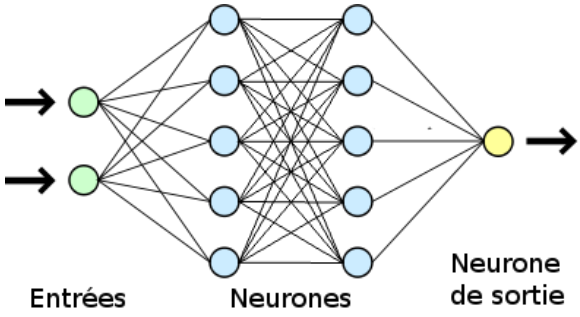
La valeur des pixels des images ci-dessus est différente de la valeur des pixels des images présentées plus haut, alors qu'elles représentent les mêmes fruits. Il existe effectivement un nombre immense d'images d'un même fruit. Mais d'une certaine façon, notre cerveau parvient à interpréter ces différentes combinaisons de pixels comme étant le même fruit et de reconnaître d'autres images comme étant des fruits différents. De cette façon, un ordinateur devra calculer une combinaison des pixels d'une image pour reconnaître ce qu'elle représente. Pour cela, imaginons que les pixels d'une image en noir et blanc ne sont en réalité que des nombres rationnels compris entre 0 et 1. Un pixel noir est représenté par le chiffre 1 et un pixel blanc par 0. Un pixel gris foncé aura par exemple une valeur de 0.9. De ce même principe, il est possible de reconnaître des images en couleur.

2 – Projet de reconnaissance de fruits

Réseau neuronal :



Un neurone est plus ou moins ce qu'on a appelé jusqu'ici une sortie, c'est le résultat de la somme des produits entre les neurones précédents (ici, les pixels) et les synapses. Un neurone peut simplement être catégorisé comme « quelque chose » qui contient un nombre. L'assemblage de plusieurs de ces neurones forment alors un réseau de neurones. Il est possible de créer différentes couches de neurones. Par exemple, pour résoudre des problèmes plus complexes comme par exemple reconnaître des visages, un simple réseau avec une seule couche de neurones ne suffira pas. Il faudra alors un réseau plus complexe avec plus de couches de neurones pour mieux reconnaître les détails.



Voici, ci-dessus, à quoi ressemble un réseau neuronal. Il prend des entrées, les neurones agissent et un neurone ressort en sortie.

A - Version 1 : Version de Test

Dans la première version de notre projet nous utilisons seulement les library numpy, panda et d'autres pour faciliter l'ouverture et la lecture des images de notre base de données.

Dans cette version, le réseau neuronale de notre IA possède 30000 entrées qui correspond au nombre total d'octet de nos images (100 * 100 pixels * 3).
Suivi de 6 couches cachées de 15 neurones chacune pour ensuite avoir une seule sortie qui correspond au type de fruit.

Apprentissage automatique :

Pour apprendre à notre IA à faire la différence entre différents fruits, nous possédons une base de données contenant de multiples types de fruits ainsi que leurs noms.

Avant de la tester avec des images prises sur internet, il faut entraîner notre IA. Pour ce faire, nous chargeons toutes les images de notre base de données dans deux tableaux.

- Le premier contiendra une liste de tous les octets d'une image que nous enverrons dans le réseau de neurones.
- Le second contiendra un nombre situé entre 0 et 1 et qui correspond à un nom de fruit. (Exemple 1 = pomme , 0.9 = poire , 0.8 = fraise…).

Une fois les images chargées, nous testons les valeurs des octets d'une image en les faisant passer dans le réseau neuronal de notre IA qui va nous retourner une valeur entre 0 et 1 qui correspond à une prédiction sur le nom du fruit envoyé. On le teste pour toutes les images de notre base et nous obtenons ainsi une liste de prédictions.

Une fois cette liste récupérée, l'IA s'entraîne en calculant la marge d'erreur entre ses prédictions et les résultats attendus grâce au second tableau que nous avons rempli juste avant. Et pour chaque résultat, en remontant dans son système neuronal, elle adapte les poids de ses neurones dans le but de sortir une prédiction plus proche de la réalité la prochaine fois.

En répétant le processus des dizaines de milliers de fois, nous obtenons une IA capable de faire des prédictions extrêmement précises.

Premier problème : la taille des images :

Notre base de données ne contient que des images de 100 * 100 pixels tandis que des images prises sur internet peuvent avoir des tailles bien plus grandes. C'est pourquoi, nous avons implémenté une fonction de redimension. Pour une image donnée, la fonction va faire une découpe de 100 * 100 pixels en respectant les proportions de l'image prise sur internet.

Une fois cette découpe faite, elle va "réduire" l'image en faisant la moyenne de tous les pixels et obtenir une image réduite de 100 * 100 pixels.

Deuxième problème : les limites de ce système :

En effet, cette version du projet comporte beaucoup de limites. Par exemple, si la taille de la base de données est trop importante, le réseau de neurones risque de ne jamais trouver de résultats corrects en tournant en rond.

Cela marche aussi si le nombre de fruits différents est trop important, le système n'arrivera pas à trouver une bonne solution.

Cette version restera donc en tant que version de test de ce projet pour comprendre le principe de l'intelligence artificielle.

B - Version 2 : Version Finale

Dans cette version, nous utiliserons bien plus de library comme **keras** qui va nous permettre de faciliter la création de modèle/ réseau de neurones plus performant que celui de la première version. Notamment grâce à la classe Keras Conv2D.

Apprentissage automatique :

Pour lui apprendre à reconnaître des fruits il faut l'entraîner, pour ce faire nous utilisons un premier tableau dans lequel nous allons spécifier le nom du fichier dans lequel **se trouve notre base de données**. Nous faisons de même pour le fichier de **test** qui va nous permettre de **tester notre IA avec des photos de fruits lambda**.

Une fois les chemins placés dans des variables nous allons lire tous les fichiers de la base de données et définir un Label pour chaque image présente à l'intérieur qui correspond au Nom du fruit sur l'image. Nous récupérons ensuite le chemin de l'image pour créer un tableau contenant les valeurs des octets d'une image.

Lorsque tout est chargé, nous entraînons notre programme avec les images en sauvegardant à chaque fois notre modèle dans un fichier pour l'utiliser une fois terminé. Après l'entraînement terminé, nous testons l'IA avec le fichier de test pour définir le Nom du fruit présent sur l'image.

3 – Conclusion

Suite à nos recherches et nos essais, nous avons créé un réseau neuronal capable de faire une rétrospection sur lui-même afin de s'adapter au résultat souhaité et ainsi améliorer ses performances. Cependant nous avons rencontré quelques problèmes avec cette version.

Nous avons donc décidé de nous orienter vers des bibliothèques plus orientées sur le type de notre projet pour réaliser une deuxième version plus performante. Mais par manque de temps nous n'avons pas pu finaliser notre projet.

En effet, pour réaliser un bon test sur notre programme, plusieurs étapes doivent être respectées. Des étapes qui peuvent être très bien adaptées sous forme de fonctions que nous n'avons pas pu implanter.

Cependant les tests que nous avons réalisés fonctionnent avec la totalité de notre base de données.