Comment une machine voit-elle ?

Vision par ordinateur et biais de données

Mise en situation. Votre téléphone se déverrouille en reconnaissant votre visage, une application range vos photos par thèmes, une voiture repère les piétons… Ces IA de **vision par ordinateur** « voient » des images.
Pourtant, une image n'est pour elles qu'un tableau de nombres. Et une IA de vision n'apprend que ce que lui montrent ses images d'entraînement : si ces images sont mal choisies, elle apprend… de mauvaises habitudes. On parle alors de **biais de données**.
Dans cette activité, vous allez découvrir comment une machine repère les contours d'un objet, puis comment des données mal choisies peuvent la tromper.

1. Repérer les contours d'un objet

Définition

Une image en niveaux de gris est un tableau de **pixels**. Chaque pixel est codé par un nombre entre 0 (noir) et 255 (blanc).
Pour repérer les **contours** (les bords des objets), on applique un **filtre** : on remplace chaque pixel par la **différence** entre son voisin de droite et lui-même. Si deux pixels voisins ont presque la même couleur, la différence est proche de 0 ; s'ils sont très différents, on est sur un bord.

Exemple

Voici une image de 5 lignes et 6 colonnes : un carré clair sur un fond sombre.

ligne 1	20	20	20	20	20	20
ligne 2	20	200	200	200	20	20
ligne 3	20	200	200	200	20	20
ligne 4	20	200	200	200	20	20
ligne 5	20	20	20	20	20	20

Sur la ligne 2, le premier pixel vaut 20 et son voisin de droite vaut 200 : la différence vaut $200 - 20 = 180$. C'est un bord : on passe du sombre au clair.

▶ À vous de jouer !

1. Calculer les 5 différences de la ligne 1 (« pixel de droite moins pixel de gauche »), puis celles de la ligne 2.
2. Où se trouvent les grandes valeurs ? Que signifie une différence positive ? négative ? nulle ?
3. Pour afficher les contours, on prend la **valeur absolue** de chaque différence, et on l'utilise comme niveau de gris. Dessiner l'image obtenue (5 lignes et 5 colonnes). Que voit-on ?
4. Ce filtre repère-t-il les bords du haut et du bas du carré ? Quel filtre faudrait-il utiliser pour les trouver ?

Coup de pouce

La valeur absolue d'un nombre est sa distance à 0 : $|-180| = 180$ et $|180| = 180$.

Remarque

Les réseaux de neurones qui reconnaissent les images (appelés réseaux **convolutifs**) appliquent des centaines de filtres de ce genre, couche après couche : les premières couches repèrent des contours, les suivantes des formes (yeux, roues…), les dernières des objets entiers. Les filtres ne sont pas choisis par un humain : ils sont appris à partir des images d'entraînement.

2. Une IA qui regarde… le décor

On veut une IA qui distingue un **loup** d'un **husky** (un chien qui ressemble beaucoup au loup). Sur chaque photo, la machine mesure deux nombres : **B** : la proportion de pixels très clairs dans l'image (en %) ; **M** : la longueur du museau, notée de 0 à 10.

Photo	1	2	3	4	5	6	7	8
Animal	loup	loup	loup	loup	husky	husky	husky	husky
Décor	neige	neige	neige	neige	herbe	herbe	herbe	herbe
B (%)	72	80	65	77	12	20	8	15
M	7	8	5	6	4	6	3	5

L'IA cherche une règle simple de la forme « si la mesure dépasse un seuil, alors c'est un loup », et garde celle qui fait le moins d'erreurs sur ces 8 photos.

▶ À vous de jouer !

1. On essaie la règle « si B > 40, alors loup ; sinon husky ». Combien d'erreurs fait-elle sur les 8 photos ?
2. On essaie la règle « si M > 5,5, alors loup ; sinon husky ». Combien d'erreurs fait-elle ?
3. Quelle règle l'IA choisit-elle ?
4. On teste les deux règles sur quatre nouvelles photos :

Photo	T1	T2	T3	T4
Animal	husky	loup	husky	loup
Décor	neige	herbe	herbe	neige
B (%)	75	10	18	70
M	4	8	3	7

   Quel est le pourcentage de bonnes réponses de chaque règle sur ces quatre photos ?
5. Qu'a vraiment appris l'IA ? D'où vient le problème ? Comment corriger les données d'entraînement ?

Coup de pouce

Une erreur, c'est un loup classé husky ou un husky classé loup. Pour la photo 3, M vaut 5, et 5 n'est pas plus grand que 5,5 : la règle répond husky, alors que c'est un loup.

Remarque

Cette expérience s'inspire d'un travail de chercheurs de l'université de Washington (2016). Ils avaient volontairement entraîné une IA avec des photos de loups dans la neige et de huskies sans neige : leur outil d'explication a montré que l'IA regardait surtout… le fond de l'image.

3. Des données qui ne représentent pas tout le monde

Définition

Une IA est **biaisée** lorsqu'elle fonctionne nettement moins bien pour certains groupes de personnes ou certaines situations. C'est souvent le cas lorsque ces groupes étaient **peu représentés** dans les données d'entraînement.

Un détecteur de visages a été entraîné avec 1 000 photos : 900 photos de personnes du groupe A et 100 photos de personnes du groupe B. On le teste ensuite sur de nouvelles photos :

	Visages à détecter	Visages détectés
Groupe A	500	490
Groupe B	100	70

▶ À vous de jouer !

1. Quelle est la proportion de photos du groupe B dans les données d'entraînement ?
2. Calculer le taux de détection global, sur les 600 visages de test. Arrondir au dixième de pourcentage. Ce détecteur semble-t-il bon ?
3. Calculer le taux de détection pour le groupe A, puis pour le groupe B.
4. Calculer le taux d'erreur de chaque groupe. Combien de fois plus d'erreurs le détecteur fait-il pour le groupe B ?
5. Pourquoi le taux global ne suffit-il pas pour juger une IA ? Quelles conséquences ce biais peut-il avoir (déverrouillage d'un téléphone, contrôle d'accès, surveillance) ?
6. On ajoute 400 photos du groupe B aux données d'entraînement. Quelle est la nouvelle proportion de photos du groupe B ? Arrondir au dixième de pourcentage.

Coup de pouce

Un taux de détection se calcule ainsi : $\dfrac{\text{nombre de visages détectés}}{\text{nombre de visages à détecter}}$. Le taux d'erreur est le complément à 100 % du taux de détection.

Remarque

Ce problème a été mesuré sur de vrais logiciels. En 2018, les chercheuses Joy Buolamwini et Timnit Gebru (étude *Gender Shades*) ont testé des systèmes commerciaux qui devinaient le genre d'une personne d'après sa photo : le taux d'erreur atteignait 0,8 % au plus pour les hommes à la peau claire, mais jusqu'à 34,7 % pour les femmes à la peau foncée. Après la publication de l'étude, plusieurs de ces entreprises ont amélioré leurs systèmes.

4. Programmer le filtre en Python

Exemple

Une ligne de pixels peut être stockée dans une liste. La fonction suivante calcule les différences entre voisins :

```
def contour(ligne):
    resultat = []
    for i in range(len(ligne) - 1):
        resultat.append(ligne[i+1] - ligne[i])
    return resultat

print(contour([20, 200, 200, 200, 20, 20]))
```

L'instruction `resultat.append(x)` ajoute la valeur `x` à la fin de la liste `resultat`.

▶ À vous de jouer !

1. Qu'affiche ce programme ? Comparer avec la ligne 2 de la partie 1.
2. Qu'affiche `print(contour([20, 20, 200, 200, 30]))` ?
3. Pourquoi la liste renvoyée contient-elle un élément de moins que la liste `ligne` ?
4. Écrire une fonction `taux(detectes, total)` qui renvoie le taux de détection en pourcentage, puis l'utiliser pour les deux groupes de la partie 3.

Coup de pouce

Pour la question 4, un pourcentage se calcule avec `detectes / total * 100`.

5. Pour aller plus loin

Remarque

Les biais de données ne viennent pas seulement des personnes représentées : l'éclairage, l'appareil photo, le pays où les photos ont été prises… peuvent aussi tromper une IA. Pour limiter ces biais, il faut des données **variées**, mesurer les résultats **groupe par groupe**, et utiliser des outils qui montrent **quelle partie de l'image** a servi à la décision.

★ Question étoilée

Dans un vrai réseau convolutif, un filtre ne regarde pas seulement deux pixels, mais un carré de 3×3 pixels. On multiplie chaque pixel par le nombre correspondant du filtre, puis on additionne les 9 produits. On utilise le filtre F ci-dessous.

Filtre F

-1	0	1
-1	0	1
-1	0	1

Carré P

20	200	200
20	200	200
20	200	200

Carré Q

200	200	200
200	200	200
200	200	200

1. Calculer le résultat du filtre F appliqué au carré P, puis au carré Q.
2. Quel type de contour ce filtre repère-t-il ?

À retenir

- Pour une IA, une image est un tableau de nombres. Des **filtres** (des calculs sur les pixels voisins) font apparaître les contours, puis les formes et les objets.
- Une IA n'apprend que ce que montrent ses données : si les données d'entraînement sont mal choisies, elle peut s'appuyer sur un détail sans rapport (le décor) ou mal fonctionner pour un groupe peu représenté. C'est un **biais de données**.
- Un bon taux de réussite global peut cacher de mauvais résultats pour un groupe : il faut mesurer la réussite **groupe par groupe** et utiliser des données **variées**.