



IA – reconnaissance d'image

Fabien Lahoudère
Matthias Spisser



Sommaire

CONTEXTE

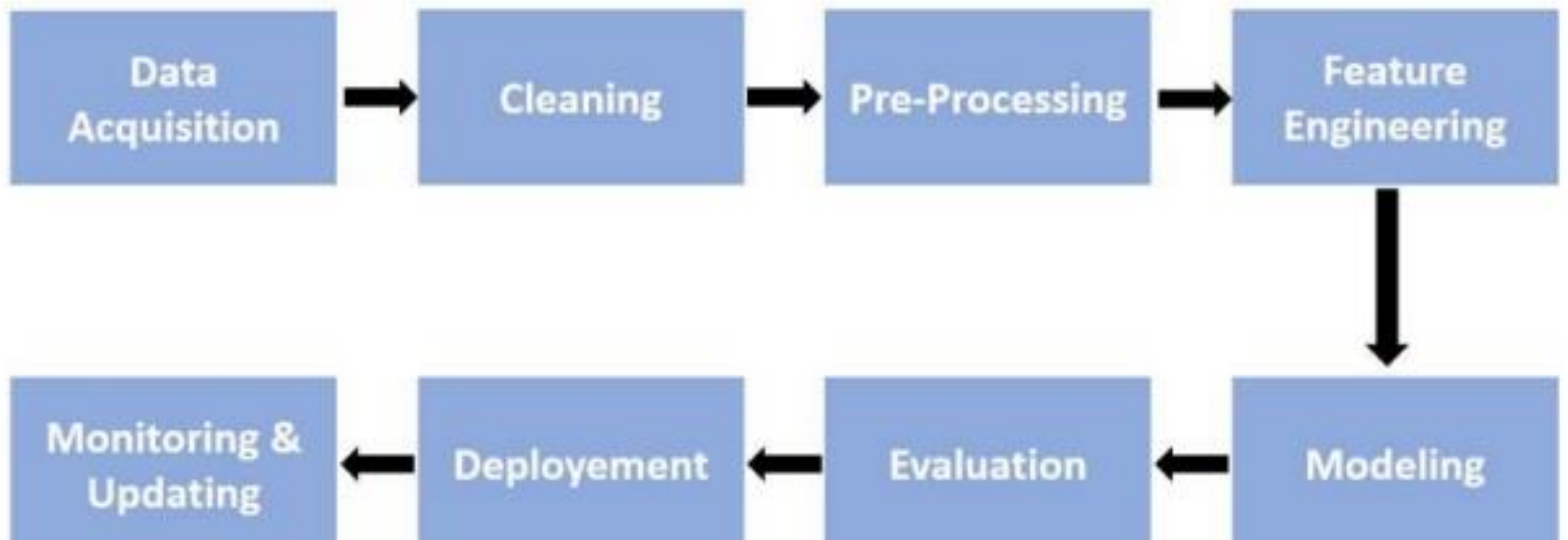
AUGMENTATION
DE DONNEES

CNN

RESULTATS

CONCLUSION

Objectifs



Chemin des données –Acquisition des données

Acquisition des données

Selenium



- Utiliser un algorithme qui automatise la recherche et la récupération de données
- Il permet d'interagir avec des navigateurs web comme google chrome ou Mozilla Firefox en utilisant les drivers web, comme le ferait un utilisateur

Acquisition des données



friche148.png



friche149.png

Mots clés : terrain pas cher
terrain paca

→ Nettoyage

Acquisition des données

Nous récupérons les images dans un dossier créé par notre algorithme.
Chaque image est identifiée par un numéro



depot25.png



depot26.png



depot27.png



depot35.png



depot36.png



depot37.png

Les 5 V en big data

Volumétrie

les volumes de données à collecter et analyser sont considérables et en augmentation constante

Variabilité

les données peuvent prendre des formes très variées et très hétérogènes

Vélocité

les données doivent être collectées et traitées en temps réel

Véracité

la véracité ou fiabilité des données

Valeur

la capacité de se concentrer sur les données ayant une réelle valeur et étant actionnables

Augmentation de données

Bibliothèque Imgaug



depot73.png



depot74.png



depot75.png



depot83.png



depot84.png



depot85.png

Bruitage, rotation, contraste

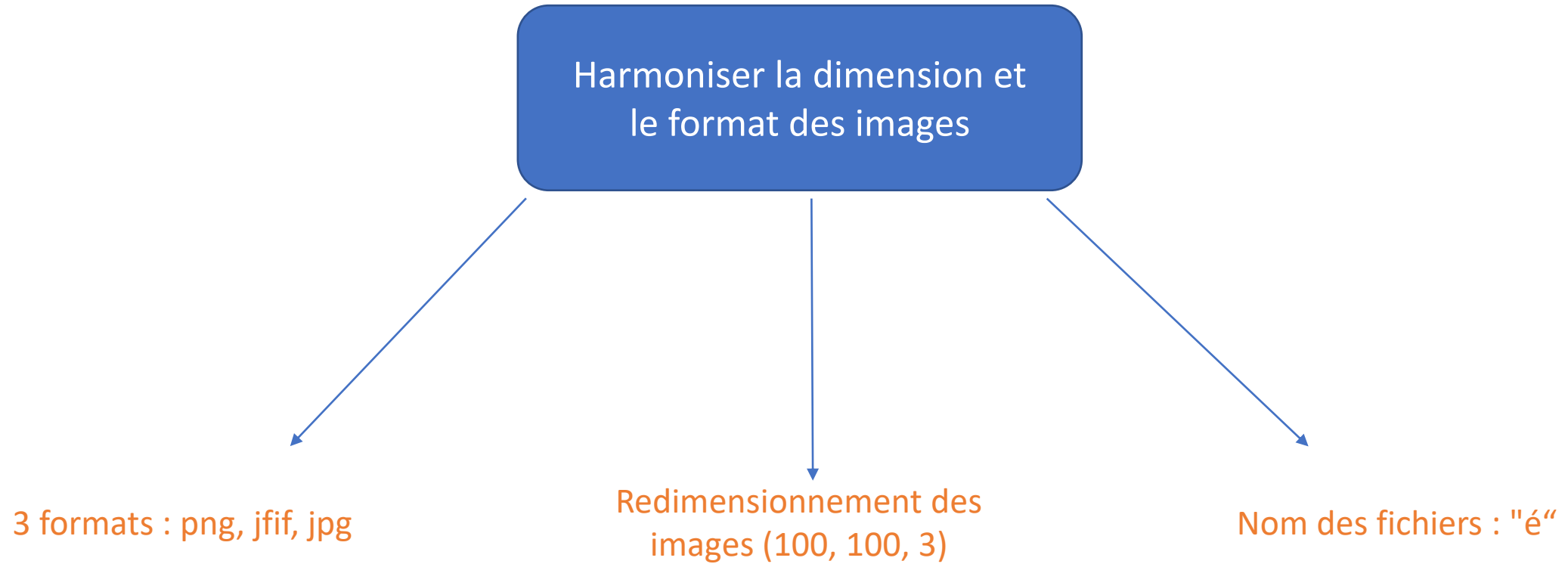
Exécuter dans un ordre aléatoire

Augmentation de données



Exemples Imgaug : ajout d'un bruit gaussien

Augmentation de données



Librairie OS



OS Module

```
>>> import os
```

Effectuer des opérations courantes : créer des dossiers, gérer des fichiers, obtenir des informations sur les fichiers, déplacer, renommer

Augmentation de données

Problème avec PNG

```
... depot1.png n'est pas un fichier PNG valide.  
    depot35.png n'est pas un fichier PNG valide.  
    depot6.png n'est pas un fichier PNG valide.  
    depot17.png n'est pas un fichier PNG valide.  
    depot25.png n'est pas un fichier PNG valide.  
    depot34.png n'est pas un fichier PNG valide.  
    depot19.png n'est pas un fichier PNG valide.  
    depot7.png n'est pas un fichier PNG valide.  
    depot8.png n'est pas un fichier PNG valide.  
    depot43.png n'est pas un fichier PNG valide.
```

¼ de nos images sont corrompues: problème réglé en redimensionnant les images

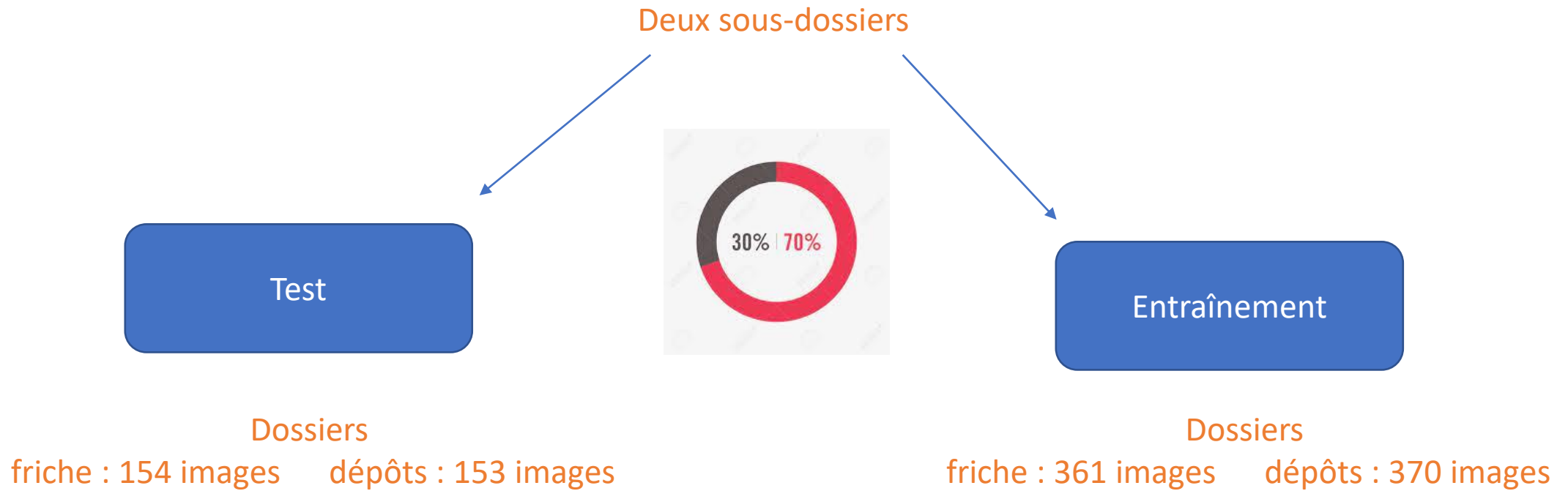
Classification

523 images de dépôts et 515 images de friche, soit 1038 images au total.

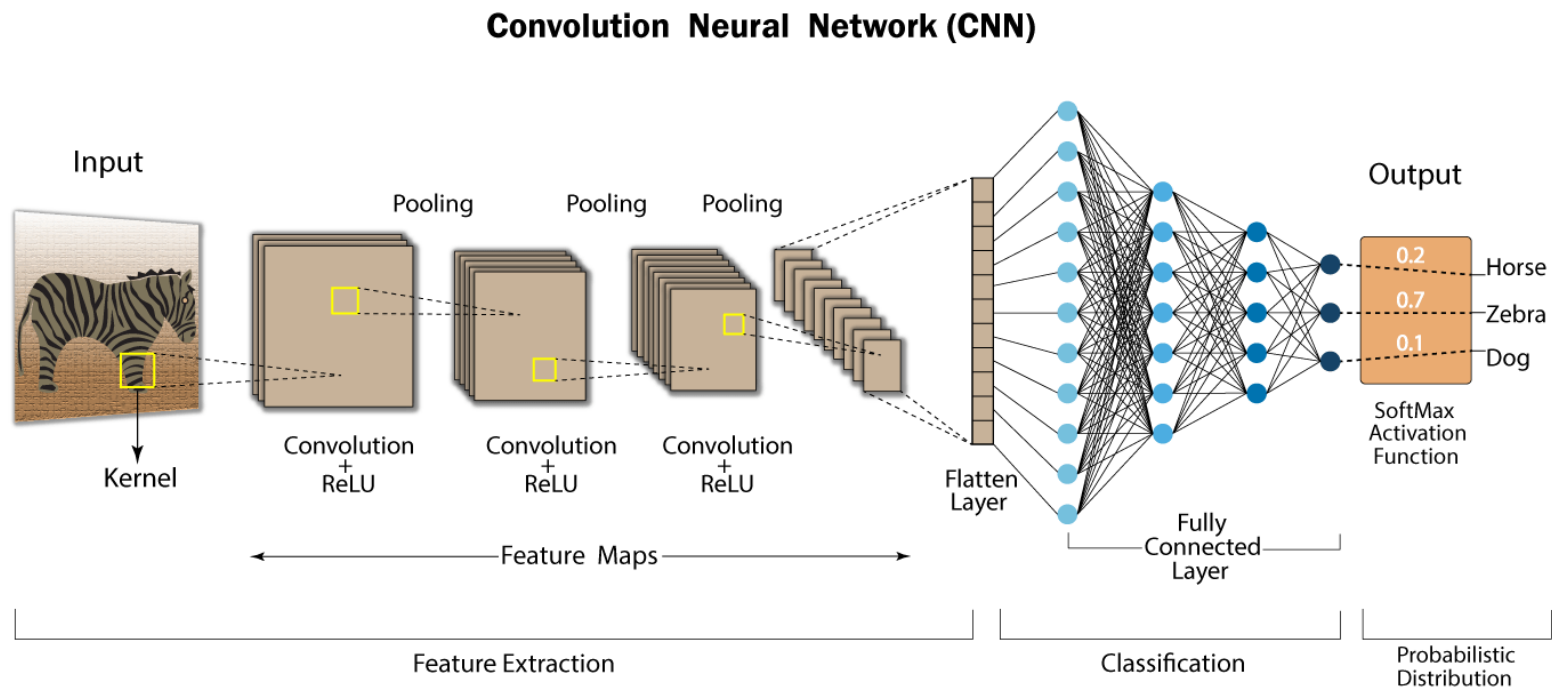
argements > database_origin

Nom	Modifié le	Type	Taille
test	15/03/2023 11:22	Dossier de fichiers	
train	15/03/2023 11:22	Dossier de fichiers	

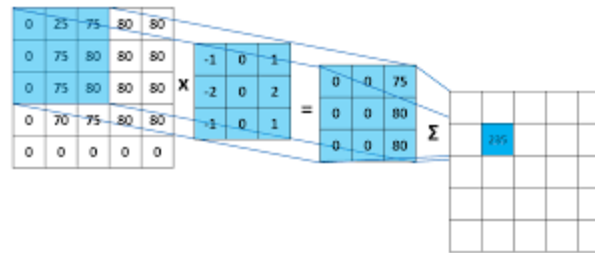
Classification



Entraînement du CNN - rappels



Entraînement du CNN - convolution



Le noyau parcourt l'image selon une direction pour faire ressortir certains détails

Cette opération permet de faire ressortir des détails extraits sous formes de caractéristiques

Axiome central: les pixels voisins sont fortement corrélés

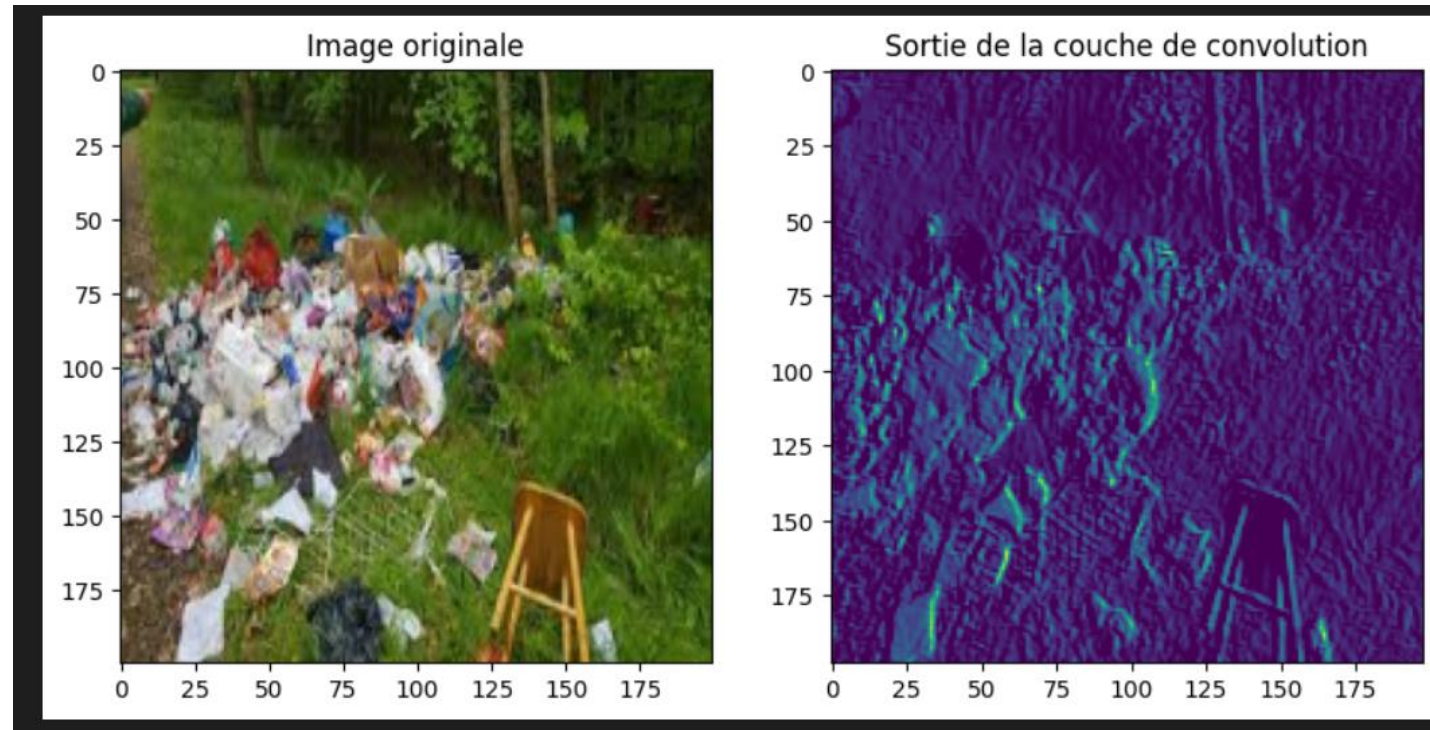
Entraînement du CNN - convolution

[200, 200, 3]

[198, 198, 32]

 Keras

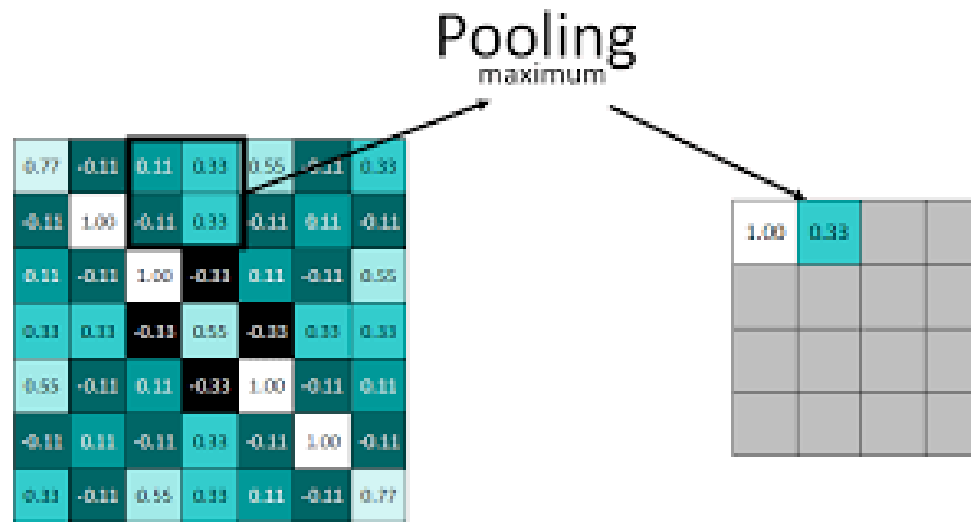

TensorFlow



Les caractéristiques les + discriminants ressortent /
Les détails les moins discriminants sont écrasés

Entraînement du CNN - couche de pooling

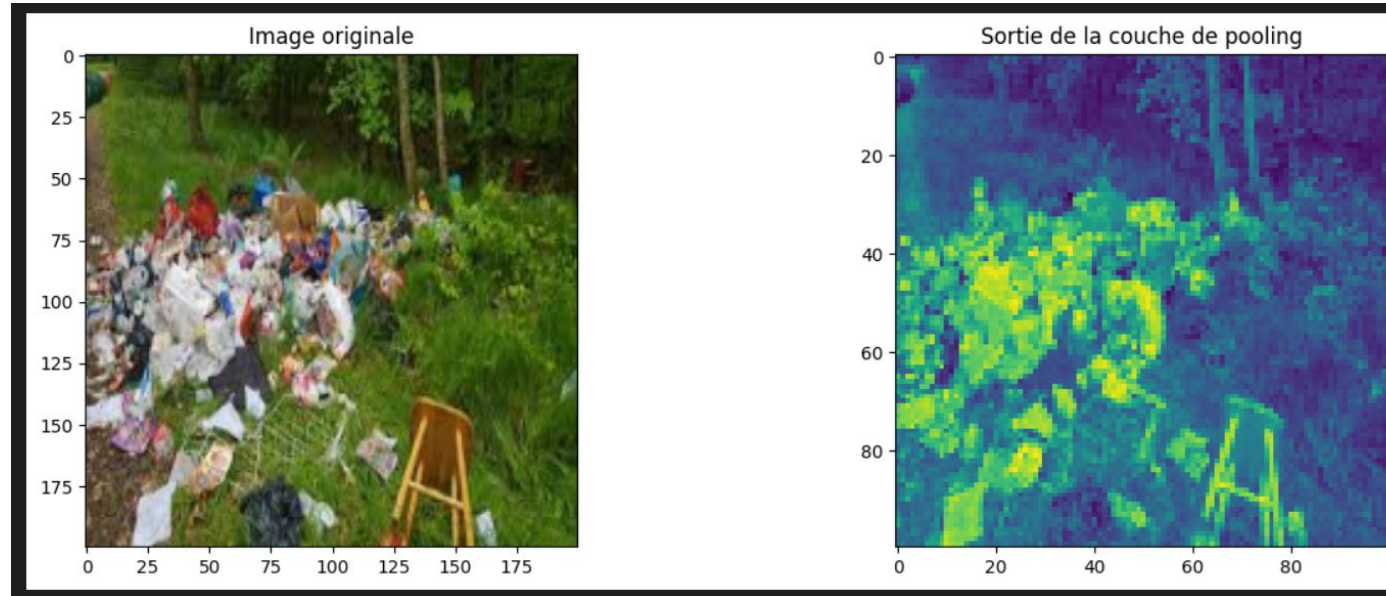
Prend les features
map en entrée



Augmenter les performances
en conservant l'intégrité des
images

Diminue la taille en
conservant un max
de caractéristiques

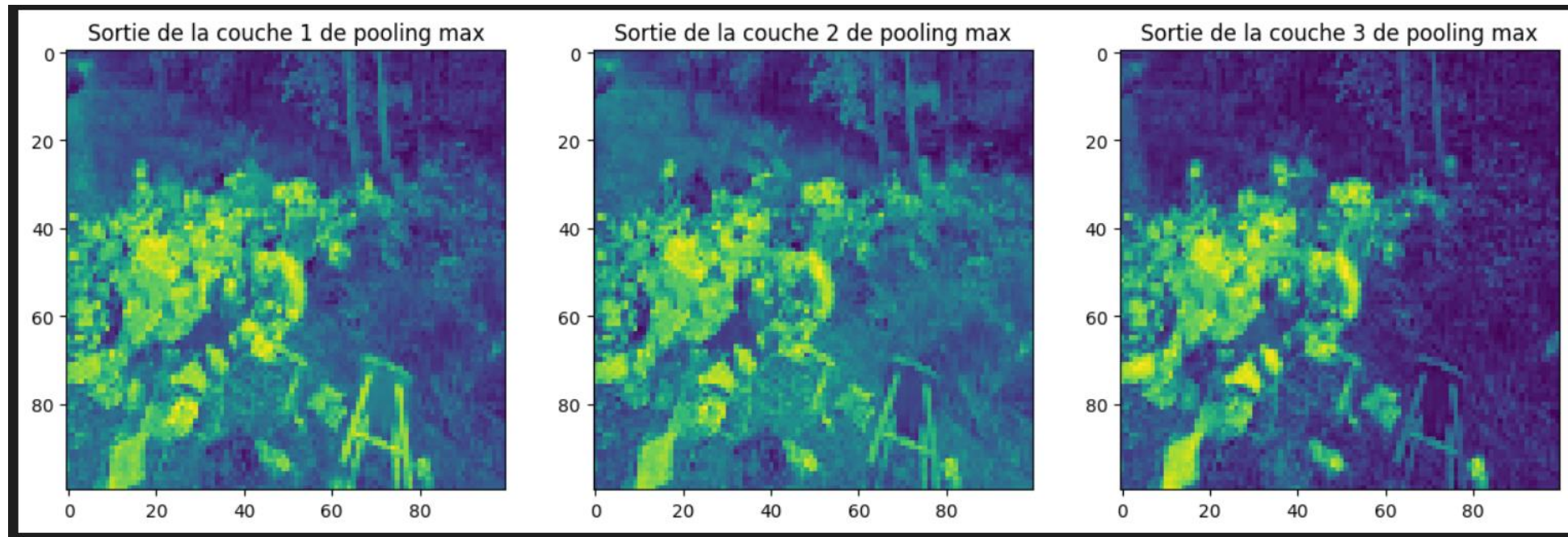
Entraînement du CNN - couche de pooling



La qualité est altérée mais
certains détails ressortent
clairement

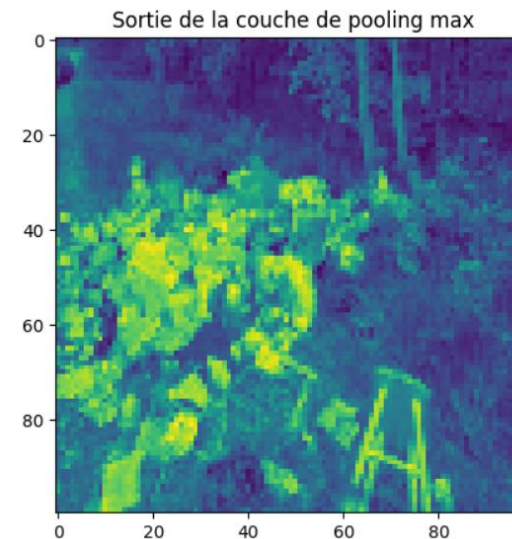
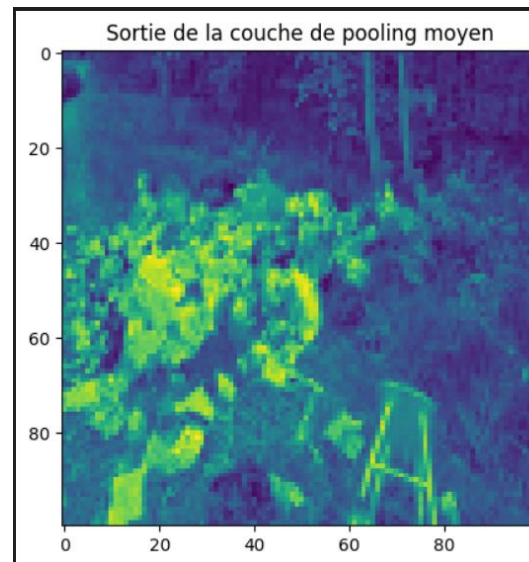
A force de répétition,
l'identification répétée de
ces features renforcera le
modèle

Entraînement du CNN - couche de pooling



Entraînement du CNN - pooling moyen ou max ?

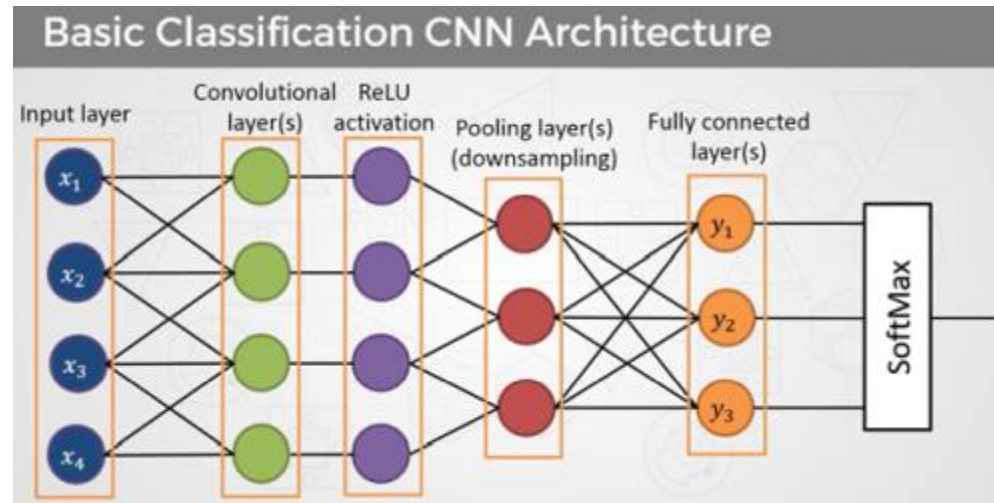
Fait ressortir les
features de manière
plus douce



Méthode plus efficace pour
identifier les détails subtils

Entraînement du CNN – couche dense

Prend en entrée les données écrasées de la couche d'aplatissement



Connecte complètement chaque neurone aux neurones de la couche dense

Effectue la classification

Entraînement du CNN – premier essai

Premier essai:
4 couches de convolution
4 couches de pooling
1 couche d'aplatissement
2 couches denses
Perte: entropie croisée

```
loss: 0.6932 - accuracy: 0.5000  
loss: 0.6932 - accuracy: 0.5000  
loss: 0.6932 - accuracy: 0.5000  
loss: 0.6932 - accuracy: 0.5000  
loss: 0.6932 - accuracy: 0.5000  
loss: 0.6932 - accuracy: 0.5000
```



Convergence à 0,5:
surapprentissage !

Pas de panique : régularisons

Désactive certaines
couches du réseau de
manière aléatoire

DROPOUT

Permet une généralisation
plus efficace du modèle

Le réseau s'entraîne en
utilisant plusieurs
combinaisons possibles de
couches du réseau

Régularisation par transfert

Réutiliser les poids d'un modèle pré entraîné pour les injecter dans notre apprentissage

APPRENTISSAGE PAR TRANSFERT

Couramment utilisé en NLP dans le traitement des langages écrit

Permet à notre algorithme de DL de s'entraîner avec moins de données

Régularisation par transfer learning : VGG16

On utilise le modèle
VGG16 entraîné sur IMG
NET



Implémentable avec
tensorflow et keras

IMGNET: 14 millions
d'images de + 1000 classes
entraînées pendant
plusieurs semaines

Régularisation par transfert : VGG16

On entraîne nos images à travers 2 couches denses et une couche d'aplatissement avec un dropout

```
loss: 0.5922 - accuracy: 0.7809 - val_loss: 0.5153 - val_accuracy: 0.8200
loss: 0.3267 - accuracy: 0.8825 - val_loss: 0.3679 - val_accuracy: 0.8100
loss: 0.3326 - accuracy: 0.8750 - val_loss: 0.6009 - val_accuracy: 0.7500
loss: 0.2525 - accuracy: 0.9068 - val_loss: 0.3661 - val_accuracy: 0.8450
loss: 0.2829 - accuracy: 0.8925 - val_loss: 0.3383 - val_accuracy: 0.8800
loss: 0.2785 - accuracy: 0.8992 - val_loss: 0.3312 - val_accuracy: 0.8900
loss: 0.2569 - accuracy: 0.9200 - val_loss: 0.2974 - val_accuracy: 0.8800
loss: 0.2115 - accuracy: 0.9270 - val_loss: 0.2503 - val_accuracy: 0.9050
loss: 0.1939 - accuracy: 0.9194 - val_loss: 0.3116 - val_accuracy: 0.8900
loss: 0.2163 - accuracy: 0.9100 - val_loss: 0.3565 - val_accuracy: 0.8900
loss: 0.2157 - accuracy: 0.9270 - val_loss: 0.3888 - val_accuracy: 0.8950
```

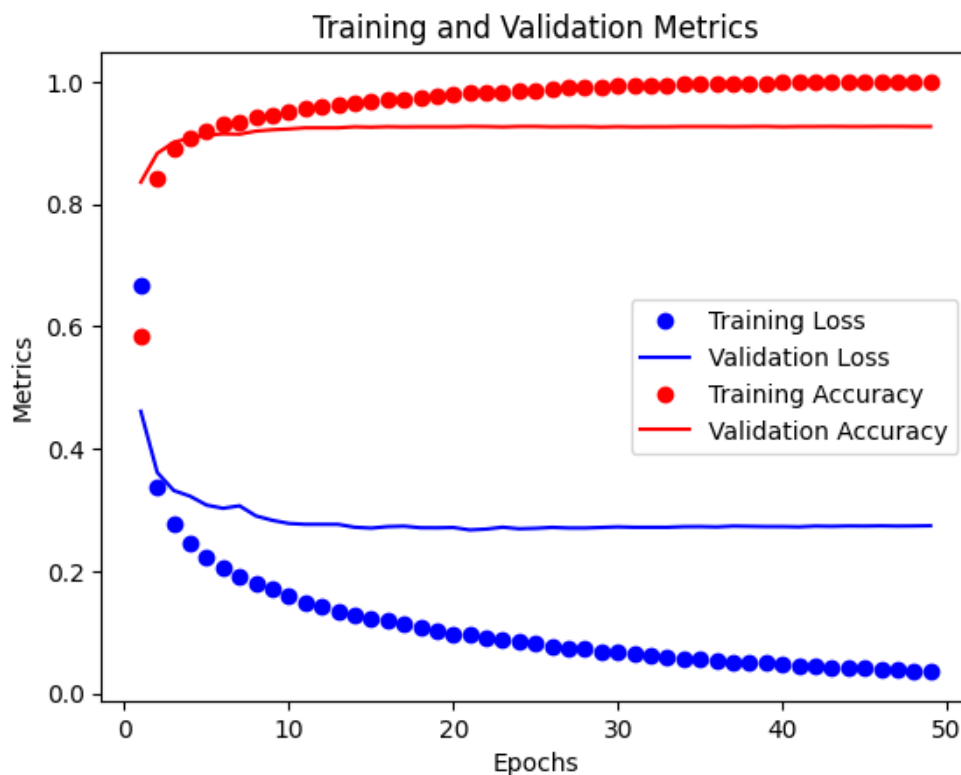
Activation relu et softmax, on entraîne sur 50 epochs

Une convergence rapide >0,8

Régularisation par transfert : VGG16



Perte: 0,0305
Précision: 0,9924



Val_loss: 0,3
Val_accuracy: 0,90

L'écart entre les valeurs de loss et de val_loss et de l'accuracy et de val_accuracy laissent présager un léger surentrainement

Détection temps réel

On enregistre une vidéo
avec 20 nouvelles images
avec apowersoft recorder



Notre code enregistre la
vidéo avec les classes
détectées sous un fichier avi

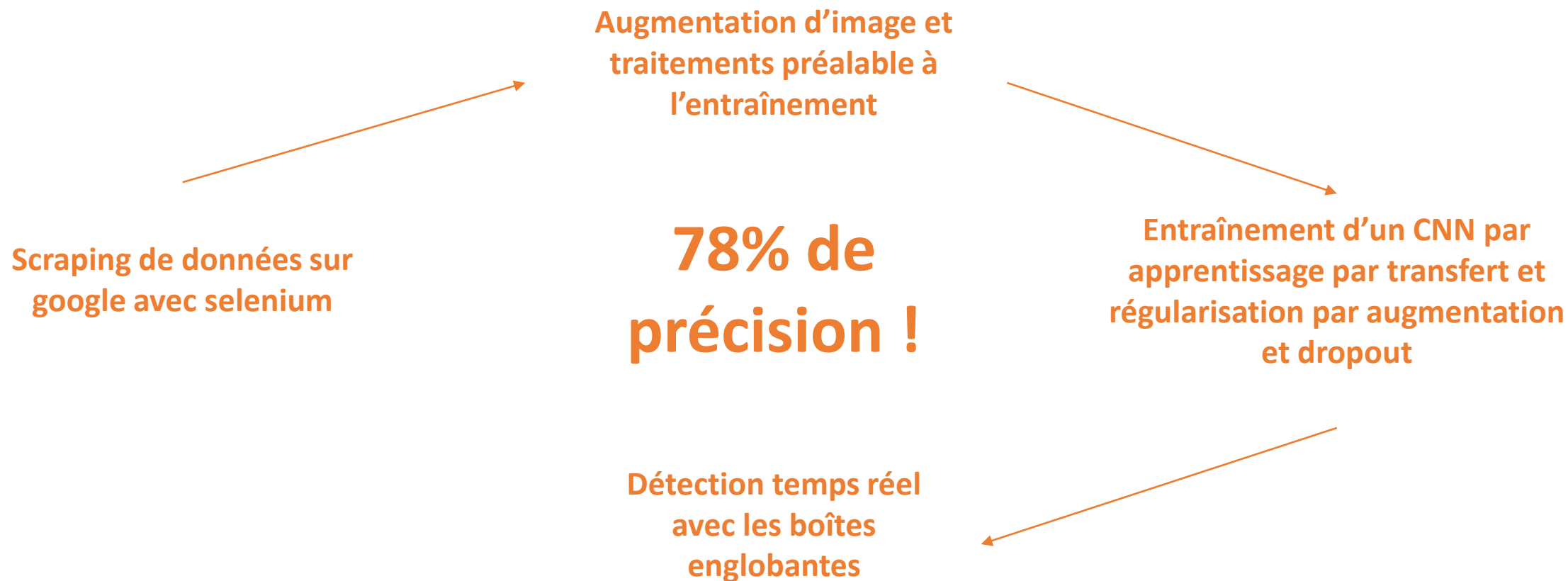
On charge le modèle et
on boucle sur les images
de la vidéo


Détection temps réel

depots: 1.00



Détection temps réel





**Merci pour votre attention !
Avez-vous des questions ?**