

# Projet 8 : Développer une preuve de concept

**Réalisé par : Taher Haggui**

# Plan:

Introduction

1

Thématique choisie

2

Domaine de détection  
d'objets et son état de l'art.

3

Jeu de données choisie et  
méthode de baseline.

4

Nouvelle approche .

5

Comparaison &  
interprétation des résultats.

Conclusion

# Problématique:



**Trouver une méthode plus récente pour améliorer la performance d'une méthode, appliqué sur un jeu de données d'un des projets de parcours , ou un autre jeu de données et une problématique de choix.**

# Comment ?

**1.Présenter une thématique et l'état de l'art.**

**2.Présenter le jeu de données choisie et la méthode baseline.**

**3.Mise en oeuvre de la nouvelle approche à implémenter .**

**4.Comparaison et analyse des résultats.**

# Résultats attendus ?

**Améliorer les performances de la méthode baseline sur le jeu de données choisi.**

# Thématique choisie :



**Détection d'objets**

# Détection d'objets : Introduction

Combien de temps avez-vous passé à chercher les clés de votre chambre perdue dans une maison désordonnée et en désordre?



**C'est la puissance des algorithmes de détection d'objets, qui pouvaient localiser vos clés en quelques millisecondes .**

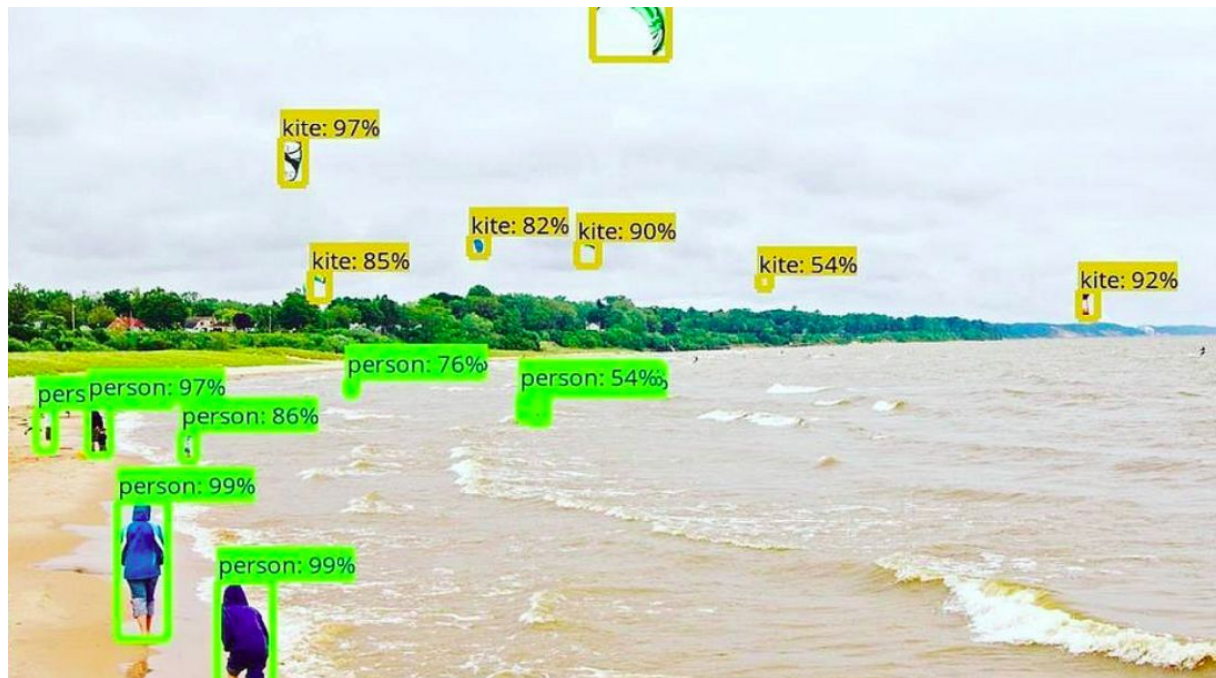
# Détection d'objets : Introduction

les domaines d'application des algorithmes de détection d'objets sont multiples , nous citons à titre d'exemple:

- Les voitures autonomes
- La surveillance 24h/24
- détection des véhicules en temps réel dans les villes intelligentes...



# Détection d'objets : Illustration



# Détection d'objets : CNN

Dans un CNN ,nous transmettons une image au réseau, qui la transmet ensuite à travers diverses convolutions et couches de pooling. Enfin, nous obtenons la sortie sous la forme de la classe de l'objet. Pour chaque image d'entrée, nous obtenons une classe correspondante en sortie.

Ainsi pour résoudre un problème de détection d'objets à l'aide d'un CNN . Nous devons procéder comme suit :

1. Premièrement, nous prenons une image en entrée:



# Détection d'objets : CNN

2. Ensuite, nous divisons l'image en différentes régions:



3. Nous considérerons ensuite chaque région comme une image distincte.

4. Transmettez toutes ces régions (images) au CNN et les classer dans différentes classes.

# Détection d'objets : CNN

5. Une fois que nous avons affecté chaque région en sa classe correspondante, nous pouvons combiner toutes ces régions pour obtenir l'image d'origine avec les objets détectés:



- ➡ Le problème avec cette approche est que les objets dans l'image peuvent avoir différents formats et différents emplacements.
- ➡ En raison de ces facteurs, nous aurions besoin d'un très grand nombre de régions, ce qui entraînerait un temps de calcul considérable.
- ➡ Donc, pour résoudre ce problème et réduire le nombre de régions, nous pouvons utiliser CNN basé sur les régions (RCNN)

# Détection d'objets : RCNN

Au lieu de travailler sur un grand nombre de régions, l'algorithme RCNN propose un ensemble de cases dans l'image et vérifie si l'une d'elles contient un objet. RCNN utilise la recherche sélective pour extraire ces zones d'une image (ces zones sont appelées régions).

Vous trouverez ci-dessous un résumé succinct des étapes suivies dans RCNN pour détecter des objets:

- ❖ Nous prenons d'abord un réseau de neurones convolutionnels pré-entraînés.
- ❖ Ensuite, ce modèle est recyclé. Nous formons la dernière couche du réseau en fonction du nombre de classes à détecter.
- ❖ La troisième étape consiste à obtenir la région d'intérêt pour chaque image. Nous modelons ensuite toutes ces régions afin qu'elles puissent correspondre à la taille d'entrée CNN.
- ❖ Après avoir obtenu les régions, nous entraînons SVM à la classification des objets et de l'arrière-plan. Pour chaque classe, nous entraînons un SVM binaire.
- ❖ Enfin, nous formons un modèle de régression linéaire pour générer des cadres de délimitation plus étroits pour chaque objet identifié dans l'image.

# Détection d'objets : RCNN

le RCNN peut être utile pour la détection d'objets. Mais cette technique a ses propres limites. L'entraînement d'un modèle RCNN est coûteuse et lente à cause des étapes ci-dessous:

- ❖ Extraction de 2 000 régions pour chaque image en utilisant la méthode “selective search”
- ❖ Extraction de features à l'aide de CNN pour chaque région d'image. Supposons que nous ayons N images, le nombre de CNN features sera alors de  $N * 2\,000$ .
- ❖ L'ensemble du processus de détection d'objet à l'aide de RCNN repose sur trois modèles:
  - CNN pour l'extraction de features.
  - Classificateur SVM linéaire pour l'identification d'objets.
  - Modèle de régression pour le resserrement des cadres de sélection.



Tous ces processus s'associent pour rendre le RCNN très lent. Il faut environ 40 à 50 secondes pour faire des prévisions pour chaque nouvelle image, ce qui rend le modèle encombrant et pratiquement impossible à construire face à un gigantesque ensemble de données.

# Etat de l'art de détection d'objets : Mask RCNN

La bonne nouvelle: nous disposons d'une autre technique de détection d'objets qui corrige la plupart des limitations observées dans RCNN . Parmi ces algorithmes , nous trouvons :

- Fast RCNN.
- Faster RCNN .
- et le state of the art Mask RCNN.



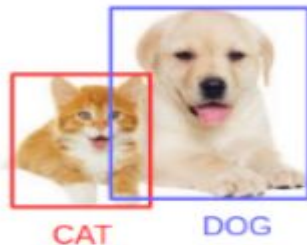
**Mask R-CNN est essentiellement une extension de Faster R-CNN**

# Etat de l'art de détection d'objets : Mask RCNN

Faster R-CNN est largement utilisé pour les tâches de détection d'objets. Pour une image donnée, elle renvoie les coordonnées de l'étiquette de classe et du cadre de sélection pour chaque objet de l'image. Alors, disons que vous transmettez l'image suivante:



Le modèle Faster R-CNN renverra quelque chose comme ceci:





# Etat de l'art de détection d'objets : Mask RCNN



**Pour une image donnée, le masque R-CNN, en plus des coordonnées de l'étiquette de classe et du cadre de sélection pour chaque objet, renverra également le masque d'objet.**

# Jeu de données choisie et méthode de baseline

## **Jeu de données:**

**Jeu de données du projet 7, qui consiste à entraîner un algorithme sur ce jeu de données et que celui-ci doit être capable de détecter la race du chien sur une photo.**

## **Méthode de baseline:**

**Réentraîner le réseau pré-entraîné VGG16.**

# Méthode de baseline: Choix du structure et paramètres d'entraînement

## Structure :

- Garder les poids de cinq premiers couches de réseau VGG16 et réentraîner le reste des autres couches de convolution et de pooling.
- Ajouter 3 couches fully connected au top de notre réseau .

## Paramètres de compilation:

- Fonction de loss : La fonction “cross entropy”.
- Optimiseur: Nous avons utilisé la fonction stochastic gradient descent (SGD) avec les paramètres suivants:
  - taux d'apprentissage: 0.001
  - momentum: 0.9

## Méthode de baseline: Résultats

**L'entraînement de convnet de baseline sur 70 % de données , pendant 600 epochs à déboucher sur les résultats suivants:**

- **Taux de précision (Accuracy) : 57%**
- **Valeur d'erreur (Loss) : 1.6**

## Nouvelle approche : Mise en oeuvre

Notre nouvelle approche qui consiste à combiner un algorithme Mask RCNN et un convnet classique dont celui de notre baseline pour résoudre notre problématique.

- Mask RCNN , pour extraire la partie intéressante de chaque image, correspond à la partie qui entoure l'objet à classer.
- Un convnet classique pour le réentraîner sur les nouvelles images affinées .

# Nouvelle approche : Mise en pratique

Pour la mise en pratique de notre approche, nous procédons tout d'abords à préparer le réseau Mask RCNN comme suit :

**étape 1: cloner le Mask RCNN repository .**

**étape 2: Télécharger les poids de réseau pré-entraîné sur MS COCO .**

**étape 3 : Configurer un modèle et faire une prédiction .**

**étape 4: Définir le model .**

**étape 5 :charger les poids de MS COCO.**

# Nouvelle approche : Mise en pratique

Le résultat de Mask RCNN préparé contient un dictionnaire pour chaque image transmise . Le dictionnaire comporte des clés pour les cadres de sélection, les masques, etc., et chaque clé pointe vers une liste de plusieurs objets possibles détectés dans l'image. les clés du dictionnaire sont les suivants:

**‘Rois’**: les boîtes ou les régions d’intérêt (ROI) liées aux objets détectés.

**‘Masques’**: masques des objets détectés.

**‘Class\_ids’**: Les entiers de classe pour les objets détectés.

**‘Scores’**: probabilité ou confiance pour chaque classe prédite.

# Nouvelle approche : Mise en pratique

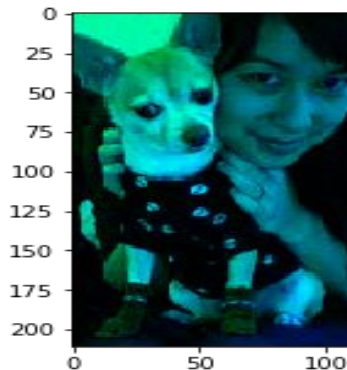


Par la suite , nous avons créé une fonction qui consiste à :

- Alimenter l'algorithme préparé Mask RCNN par l'image introduite comme paramètre
- Tester s'il y aurait un chien parmi les objets détecté.
- S'il y a un chien , la fonction extraire une nouvelle image , correspond aux pixels qui constituent la zone intéressante (ROI), fournie par Mask RCNN.
- Si non, la fonction garde la même image d'origine avec tous ses pixels



# Nouvelle approche : Mise en pratique



La prochaine étape consiste à :

- Affiner les images de notre jeu de données , moyennant la fonction utilisée.
- Réentraîner le convnet de baseline sur les nouvelles images affinées.

# Comparaison & interprétation des résultats

	Accuracy	Loss
Baseline	57.7	1.63
Nouvelle approche	53.77	1.96

# Conclusion

Dans ce projet , nous avons mis en oeuvre une nouvelle approche permettant d'extraire les régions intéressante de l'objet à classer . Cependant celle -ci n'a pas pu amélioré les performances de la méthode baseline . Ceci pourrait être expliqué par :

- ❖ Le pourcentage des images où notre algorithme Mask RCNN, n'a pas pu détecter l'objet chien .
- ❖ La deuxième raison , c'est le fait que nous sommes face à un problème de classification de race de chien , donc il serait très difficile pour notre algorithme d'extraire les features les plus distinctives de chaque race à cause de la similarité entre les races.

Ainsi nous recommandons pour pallier à ces deux problèmes de :

- Améliorer notre algorithme pour qu'il puisse détecter des objets connexes à l'objet chien tels que les objets chat , cheval , mouton dans le cas où il ne pouvait pas détecter l'objet chien .
- Faire des expérimentations de cette approche sur d'autres problèmes de classification lié à des catégories différentes.