



Université Constantine 2 -Abdelhamid Mehri  
Faculté des **Nouvelles Technologies** de l'Information et de la **Communication**  
Département Informatique **Fondamentale** et ses **Applications**



**Master Sciences et Technologies de l'Information et de la Communication STIC**

Matière : **Imagerie et Vision Artificielle ImVA**

Chapitre 4 : **La Détection des objets**

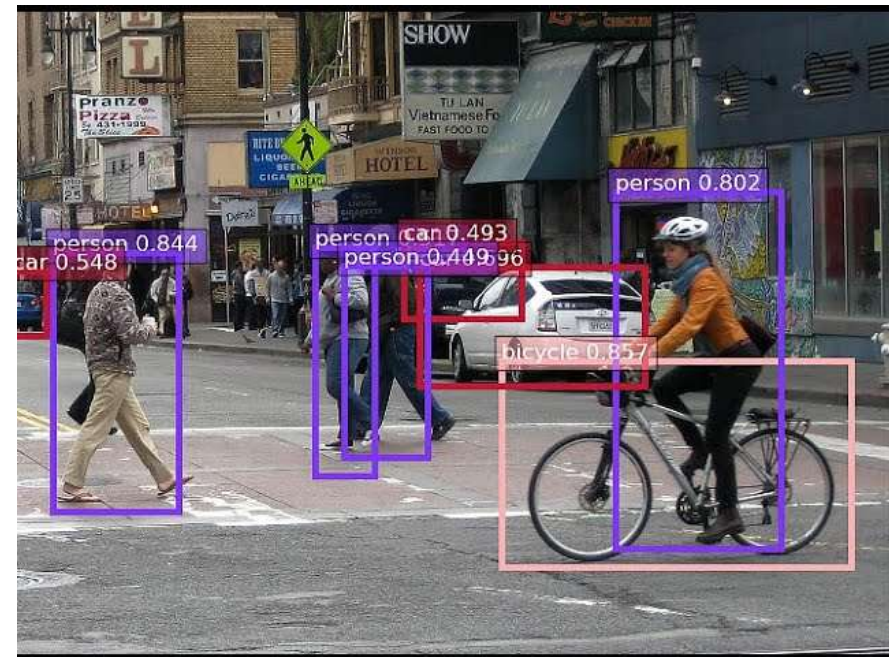
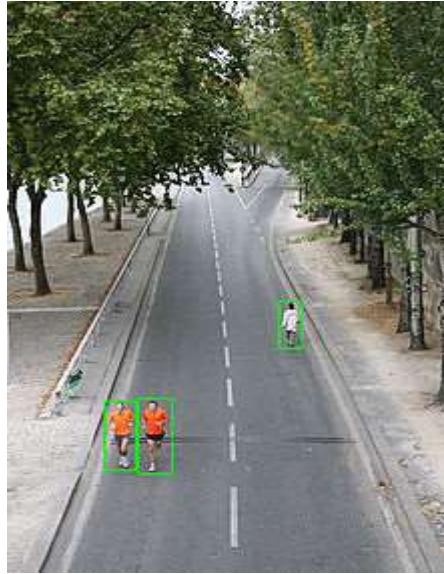
# 1- Objet

**L'objet** désigne une zone de l'image qui est caractérisé par sa texture, sa forme, sa couleur, son orientation du gradient ou encore son mouvement .

**On distingue deux types d'objets :**

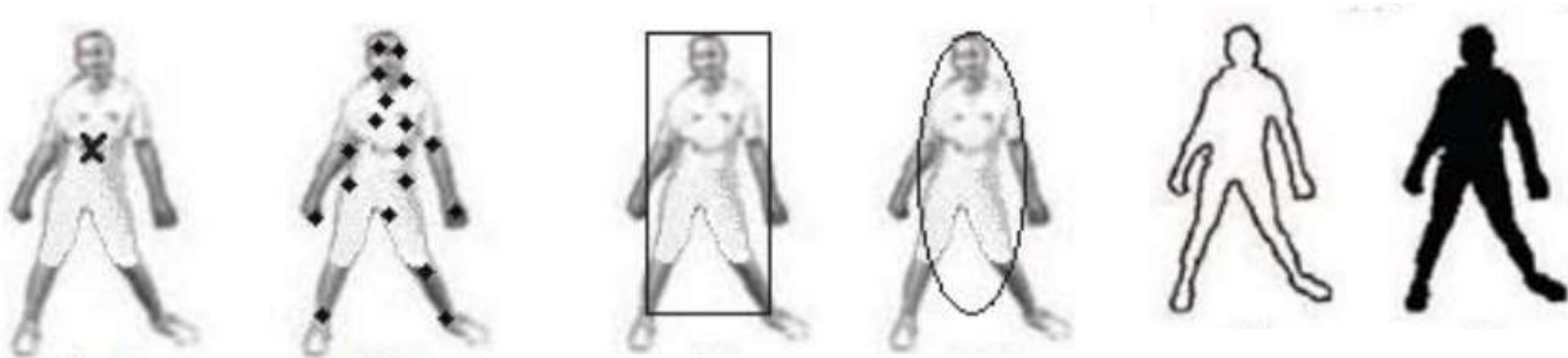
1. Les objets immobiles (stationnaires) : sont des objets qui gardent la même position d'une image à une autre tel que les arrières plans. c'est à dire sont des objets qui ne se déplacent pas au cours du temps dans la scène
2. Les objets mobiles (en mouvement) : sont des objets qui se déplacent d'une image à une autre, ils sont caractérisés par leurs mouvements dans la séquence d'images, selon leurs mouvements nous pouvons distinguer :
  - ✓ Les objets mobiles rapides : leurs déplacements est important.
  - ✓ Les objets lents : les objets qui se déplacent lentement dans la scène.
  - ✓ Les objets qui ont un mouvement simple, sont des objets qui ont le même mouvement (en vitesse et direction) dans la séquence d'image.
  - ✓ Les objets qui ont un mouvement complexe, sont des objets qui ont un mouvement aléatoire dans la séquence d'image

# 1- Objet



# 1- Objet : Représentation de la forme

- 1. Représentation par points** : l'objet est représenté par un point (centre de l'objet) ou par un ensemble de points. Ce type de représentation est généralement souhaitable pour les objets qui occupent une petite partie de l'image.
- 2. Représentation par formes géométriques primitives** : l'objet est représenté par une forme géométrique simple telle qu'un rectangle, une ellipse, etc. permettant une description de la dimension de l'objet. Cette représentation est particulièrement adaptée au suivi d'objets rigides (véhicules, ...) mais peut également convenir pour des objets non-rigides.
- 3. représentation par silhouettes et contours** : la représentation d'un objet par son contour permet de définir les limites exactes de l'objet. La région interne du contour est appelée silhouette de l'objet et peut être utilisée conjointement à l'information de contour pour le suivi d'objets. L'utilisation de cette représentation convient au suivi d'objets complexes non-rigides





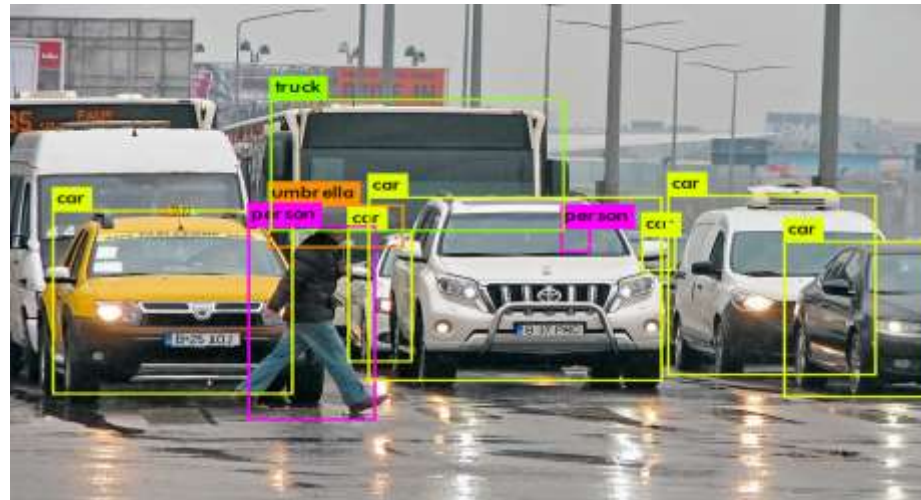
# 1- Objet

1. **Représentation par modèle articulé** : cette représentation se base sur la cinématique de l'objet. Objet articulé est composé de plusieurs parties du corps qui sont modélisées par une primitive géométrique (par exemple une ellipse) et sont reliées entre elles par des connexions. Cette représentation est adaptée au suivi d'objets articulés (suivi de personne).
2. **Représentation par squelette** : Le squelette d'un objet peut être extrait en utilisant la transformation de l'axe central. Ce modèle est couramment utilisé comme représentation de forme dans la reconnaissance d'objet. Cette représentation est souhaitable pour les objets rigides et articulés.



## 2- Détection d'objets

- **La détection d'objets** est un phénomène de vision par ordinateur qui implique la détection de divers objets dans des images ou des vidéos numériques.
- Elle consiste à reconnaître des objets dans un environnement et à les classer par catégorie.
- Les objets peuvent être de toutes formes et de toutes tailles.
- La détection d'objets peut être utilisée dans diverses applications, telles que la sécurité, la robotique, la vision par ordinateur et la reconnaissance de formes.



## 2- Détection d'objets

Ce phénomène cherche à répondre à deux questions fondamentales :

- **Qu'est-ce que l'objet ?** Cette question vise à identifier l'objet dans une image spécifique.
- **Où est-il ?** Cette question vise à trouver l'emplacement exact de l'objet dans l'image.

L'image n'est pas contrainte d'avoir un seul objet, mais peut contenir plusieurs objets. La tâche est de classer et de localiser tous les objets dans l'image.

La localisation est faite en utilisant le concept de boîte de délimitation qui peut être identifiée par certains paramètres numériques tout en respectant la limite de l'image.

# 3- Concepts de base

- La classification d'images consiste à prédire la classe d'un objet dans une image.

- Entrée : une image avec un seul objet.
- Sortie : une étiquette de classe.



CHAT

- La **localisation d'objets** consiste à identifier l'emplacement d'un ou plusieurs objets dans une image et à dessiner un cadre de délimitation autour de leur étendue.

- Entrée : une image avec un ou plusieurs objets.
- Sortie : un ou plusieurs cadres de délimitation.



- La **détection d'objets** combine ces deux tâches et dessine un cadre de délimitation autour de chaque objet dans l'image et leur attribue une classe.

- Entrée : une image avec un ou plusieurs objets.
- Sortie : un ou plusieurs cadres de délimitation et une étiquette de classe pour chaque cadre de délimitation.



CHAT



# 4- Quelques domaines d'application

- 1- Vidéosurveillance
- 2- Comptage de foule
- 3- Comptage de véhicules
- 4- Détection d'anomalies
- 5- Véhicules à conduite autonome
- 6- La Robotique



## 4- Quelques domaines d'application

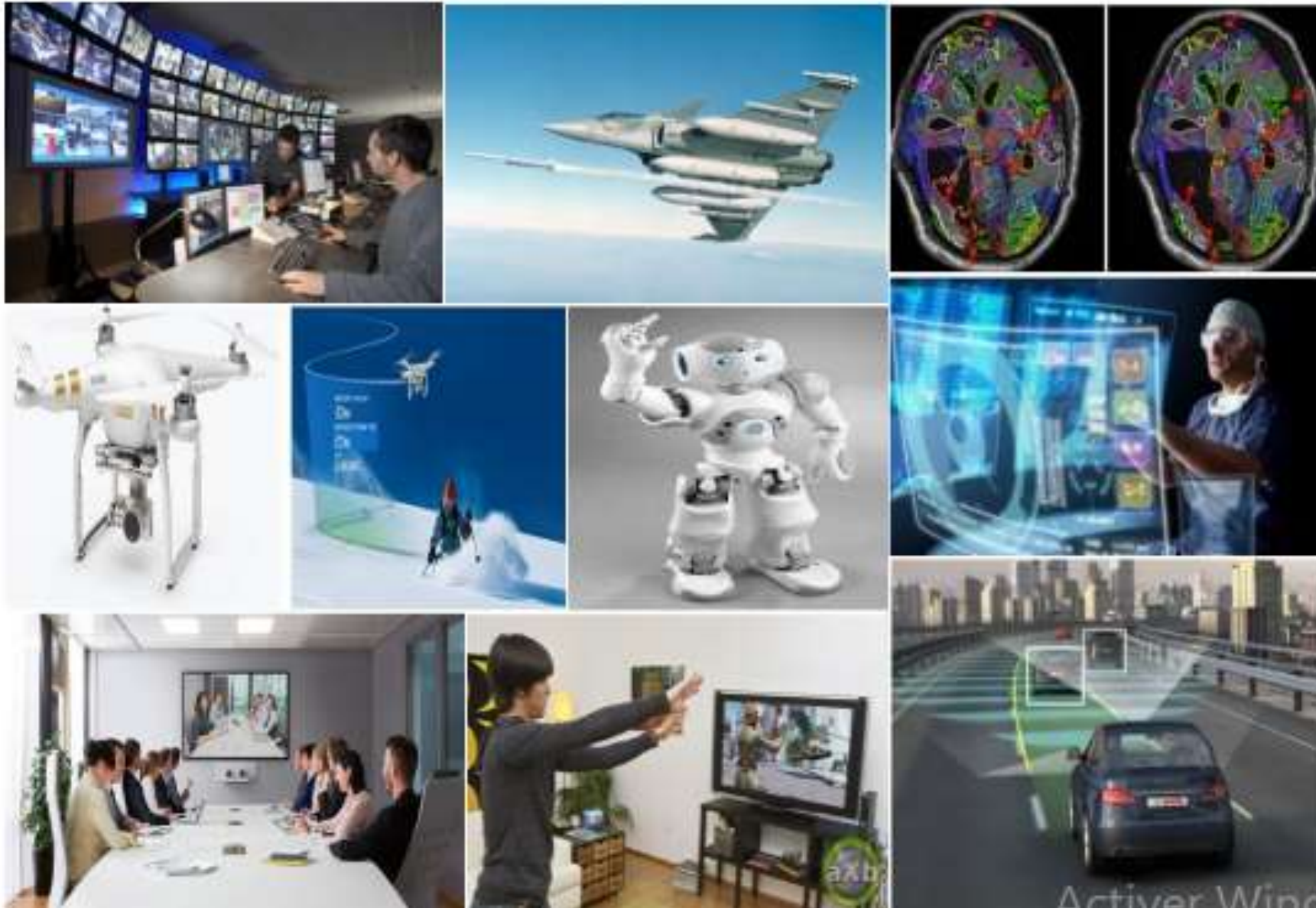




## 4- Quelques domaines d'application



## 4- Quelques domaines d'application



# 5- Les défis de la détection d'objets

Un détecteur idéal devrait avoir :

- Haute précision de localisation et de reconnaissance : le détecteur doit être capable de localiser et de reconnaître les objets dans les images avec précision.
- Haute efficacité en temps et en mémoire : la tâche de détection doit s'exécuter à une fréquence d'images suffisante avec une utilisation de mémoire et de stockage acceptable
- Plusieurs variations sont à considérer :
  - ✓ Changement d'illumination.
  - ✓ Les conditions météorologiques.
  - ✓ Changement d'échelle.
  - ✓ Occultations partielles ou totales.
  - ✓ Déformation de l'objet.
  - ✓ Objet en mouvement rapide.
  - ✓ La corruption du bruit et une mauvaise résolution.
  - ✓ Mouvement de caméra.
  - ✓ Objet de petite taille.
  - ✓ Les arrière-plans et L'ombrage.



# 5- Les défis de la détection d'objets

**Illumination**



**Occlusion**



**Deformation**



**Noise corruption**



**Out-of-plane rotation**



**Motion blurring**



# 6- Les méthodes de détection d'objets

On peut distinguer quatre ensembles d'algorithmes de détection :

1. La détection de points d'intérêt :

**Détecteur de Harris [Harris et Stephens 1988],**

**Détecteur de SIFT [Scale Invariant Feature Transform]**

**Détecteur de SURF [Speeded Up Robust Features]**

2. La détection par soustraction de fond :

**Modèles Gaussiens**

3. La segmentation d'image

**Mean Shift**

**Segmentation par Coupe-graphe**

**Contours actifs**

4. La classification supervisée.

**la méthode AdaBoost (Adaptive Boosting),**

**YOLO**

## 6- Les méthodes de détection d'objets : YOLO

Au cours des dernières années, les modèles **Deep Learning** ont gagné en renommée pour leur capacité à traiter l'information visuelle et ils sont devenus un élément clé de nombreuses applications de vision par ordinateur.

Parmi les principaux problèmes que ces modèles peuvent résoudre est la détection et la localisation des objets dans les images.

Parmi ces modèles, YOLO est l'un des plus connus et plus utilisés. Il permet de réaliser **la détection d'objets en une seule étape** et d'atteindre un très bon niveau de précision et une vitesse de détection élevée. Il est particulièrement bien adapté pour **les applications en temps réel**.

## 6- Les méthodes de détection d'objets : YOLO

### **YOLO You Only Look Once**

Les frameworks de détection antérieurs examinent différentes parties de l'image plusieurs fois à différentes échelles et ont réorienté la technique de classification des images pour détecter les objets. Cette approche est lente et inefficace.

YOLO adopte une approche totalement différente. Il **ne regarde l'image entière qu'une seule fois** et passe par le réseau une fois et détecte les objets. D'où le nom. Il est très rapide. C'est la raison pour laquelle il est devenu si populaire.

**You Only Look Once** ou YOLO est un algorithme capable de détecter les objets au premier regard, en effectuant la détection et la classification simultanément.

# Autres techniques

**Region-Based Convolutional Neural Networks (ou R-CNN)** consistaient à proposer initialement des régions d'intérêt (Rols) à partir de l'image. Pour y parvenir, on utilisait principalement des algorithmes de segmentation ou de détection des contours.

Chaque Rol était ensuite redimensionnée pour être entrée dans un classificateur tel qu'un [réseau de neurones convolutifs \(CNN\)](#). Le but était de déterminer si un objet se trouvait dans une région.

Cette technique était à l'époque une avancée significative, mais elle souffrait d'une lenteur considérable à cause du grand nombre de régions candidates à évaluer.

Pour surmonter ces limitations, **Fast R-CNN** a vu le jour. Elle permettait d'effectuer la classification des ROIs directement à partir d'une carte de caractéristiques partagées du CNN plutôt que d'utiliser des **classificateurs séparés**.

Malgré le gain de vitesse par rapport à R-CNN, le processus de sélection restait lent et complexe. C'est pourquoi l'**approche Faster R-CNN** a été créée, introduisant la proposition des régions d'intérêt par un réseau.

**Cette évolution a permis d'automatiser et d'accélérer encore davantage la génération des Rols**, mais demeurait désespérément lente à cause des multiples étapes séquentielles requises pour la détection des objets.



## 6- Les méthodes de détection d'objets : YOLO Architecture

Pour toutes ces raisons, il était grand temps qu'une nouvelle technique vienne tout chambouler dans ce domaine. Et c'est précisément ce qui s'est passé avec YOLO.

YOLO a marqué un tournant décisif, c'est à cause de son approche novatrice. En effectuant simultanément **la détection et la classification des objets** en une seule passe à travers un réseau de neurones convolutifs, elle a permis de combiner vitesse en temps réel et précision.

Son architecture en pipeline associée à des **mécanismes de régions d'intérêt** : Au lieu de proposer des régions d'intérêt dans une première étape, **elle divise l'image d'entrée en une grille de cellules**.

Chacune de ces cellules est responsable de prédire les coordonnées des **boîtes englobantes (bounding boxes)** pour les objets détectés ainsi que leurs probabilités d'appartenance à différentes classes.

Ces prédictions s'effectuent directement à partir de caractéristiques extraites par le CNN. L'utilisation d'un réseau de neurones convolutifs pour extraire les caractéristiques de l'image d'entrée est au cœur de YOLO.

Le CNN se compose de plusieurs **couches convolutives** et de **sous-échantillonnage (pooling)**, afin de capturer des motifs et des traits utiles à différentes échelles spatiales.

## 6- Les méthodes de détection d'objets : YOLO Architecture

Pour toutes ces raisons, il était grand temps qu'une nouvelle technique vienne tout chambouler dans ce domaine. Et c'est précisément ce qui s'est passé avec YOLO.

YOLO a marqué un tournant décisif, c'est à cause de son approche novatrice. En effectuant simultanément **la détection et la classification des objets** en une seule passe à travers un réseau de neurones convolutifs, elle a permis de combiner vitesse en temps réel et précision.

Son architecture en pipeline associée à des **mécanismes de régions d'intérêt** : Au lieu de proposer des régions d'intérêt dans une première étape, **elle divise l'image d'entrée en une grille de cellules**. Chacune de ces cellules est responsable de prédire les coordonnées des **boîtes englobantes (bounding boxes)** pour les objets détectés ainsi que leurs probabilités d'appartenance à différentes classes.

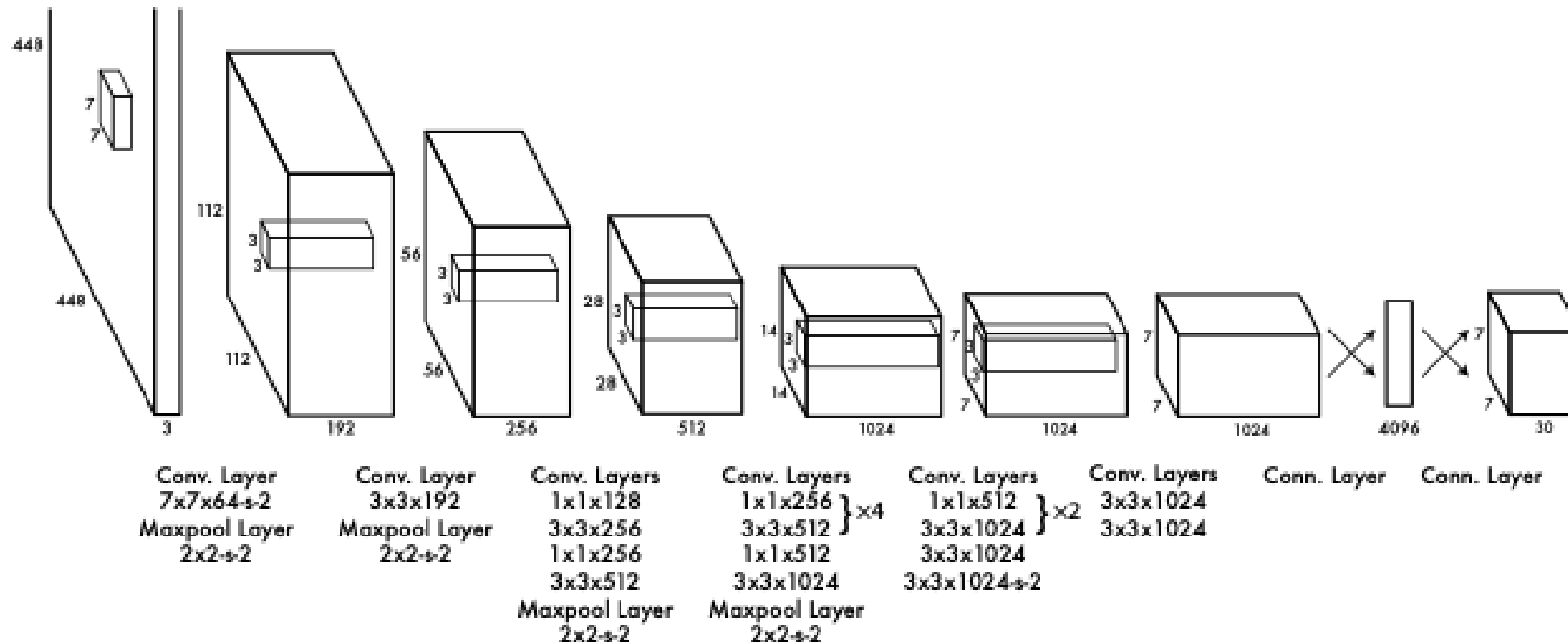
Ces prédictions s'effectuent directement à partir de caractéristiques extraites par le CNN. L'utilisation d'un réseau de neurones convolutifs pour extraire les caractéristiques de l'image d'entrée est au cœur de YOLO. Le CNN se compose de plusieurs **couches convolutives** et de **sous-échantillonnage (pooling)**, afin de capturer des motifs et des traits utiles à différentes échelles spatiales.

# 6- Les méthodes de détection d'objets : YOLO Architecture

Ceci permet d'apprendre automatiquement les représentations pertinentes des objets et d'effectuer des opérations de convolution efficacement, réduisant fortement les coûts computationnels.

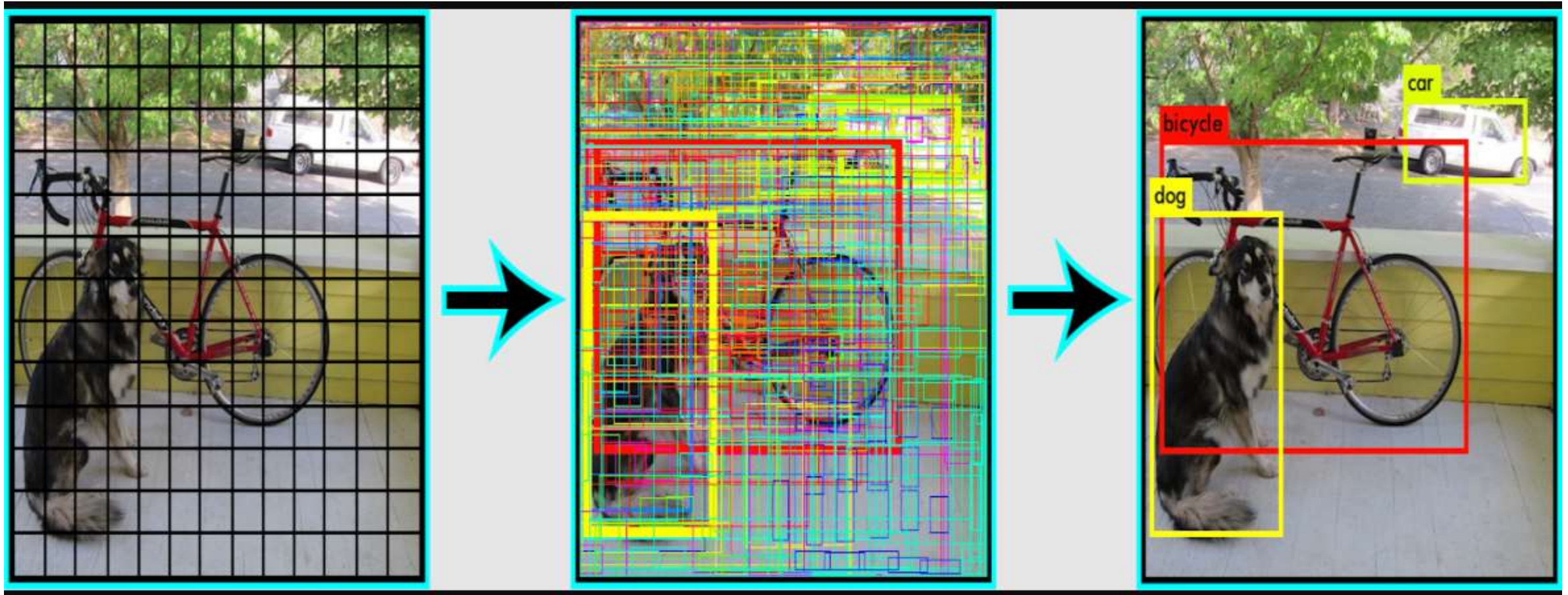
Autre caractéristique clé : l'utilisation de « **régions d'intérêt** » ou « **anchors** » qui sont des boîtes englobantes prédéfinies de tailles et formes différentes servant de référence pour les prédictions.

Chaque cellule de la grille est associée à un certain nombre d'anchors, ce qui aide YOLO à généraliser les détections sur différents types d'objets et d'échelle. Sa précision de détection s'en trouve fortement renforcée



<https://dipankarmedh1.medium.com/real-time-object-detection-with-yolo-and-webcam-enhancing-your-computer-vision-skills-861b97c78993>

## 6- Les méthodes de détection d'objets : YOLO Architecture



# Quels sont les avantages ?

**You Only Look Once** apporte **plusieurs améliorations importantes** par rapport aux précédentes méthodes de détection d'objets. C'est ce qui lui a permis de devenir l'un des algorithmes les plus utilisés pour la vision par ordinateur.

Son principal point fort est **sa capacité à détecter les objets instantanément**, et de réduire le nombre total de calculs requis. L'utilisation des ressources est optimisée puisque les caractéristiques partagées sont calculées une seule fois.

De plus, **YOLO se distingue également par ses performances et sa précision**. Son approche en pipeline lui permet de généraliser sur des objets de formes variées, le rendant robuste face à une large diversité de scénarios.

**Il excelle également pour le traitement d'images haute résolution**, puisque son architecture efficace permet de traiter des images de tailles plus importantes sans sacrifier la vitesse. C'est un réel atout pour des applications comme la détection aérienne ou satellitaire.



# Quelles sont les applications ?

On l'utilise notamment **dans les véhicules autonomes, afin d'identifier et de suivre les piétons, autres véhicules, panneaux de signalisation et tout obstacle** potentiel sur la route en temps réel. C'est ce qui permet aux systèmes de pilotage de réagir instantanément aux changements pour assurer une conduite sécurisée.

Pour la surveillance vidéo, YOLO permet de **détecter les activités suspectes, les intrus ou les objets abandonnés même au sein d'une foule**. De même, on peut s'en servir pour repérer des personnes ou des animaux perdus dans des endroits difficiles d'accès. Il peut aussi être exploité pour les contrôles de sécurité dans les aéroports, gares et autres infrastructures.

Sa capacité de reconnaissance d'activité peut aussi être utile pour **détecter et suivre les mouvements d'individus dans des séquences vidéo**. C'est un cas d'usage pertinent pour la surveillance de grands espaces tels que les stades, les centres commerciaux ou lors de grands événements sportifs.

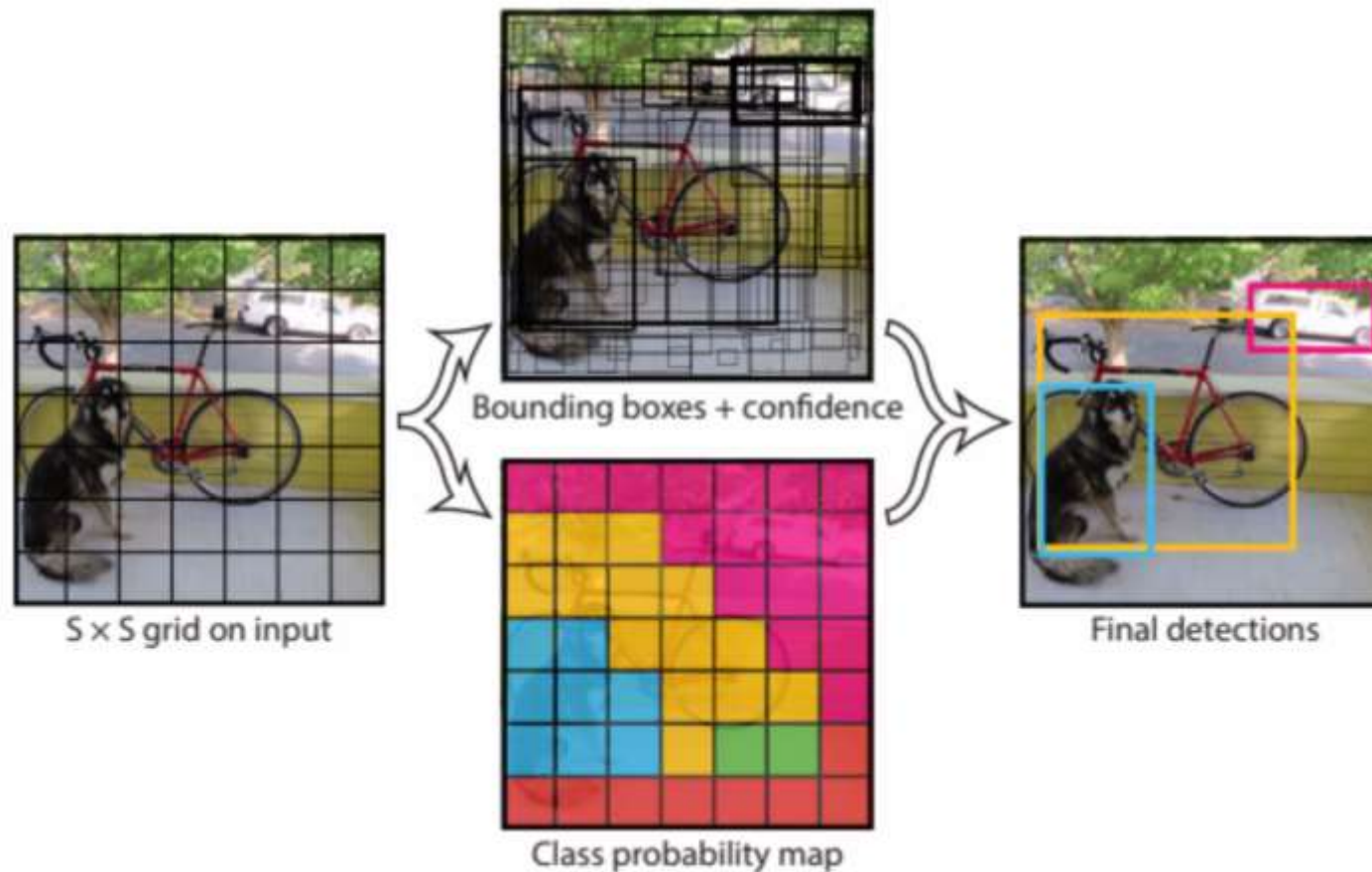
Un autre exemple d'application est la **gestion du trafic automobile**. L'algorithme peut être utilisé pour la reconnaissance automatique des plaques d'immatriculation et des visages.

Dans le domaine de la médecine, You Only Look Once est utilisé pour **la détection d'anomalies ou d'objets spécifiques dans les images médicales** telles que les radiographies, les IRM ou les scanners. Il peut contribuer à diagnostiquer de façon précoce les maladies en accélérant le processus d'analyse d'images. Ceci peut s'avérer vital dans les situations d'urgence médicale.

# Différentes Variantes De YOLO

## 1. YOLOv1

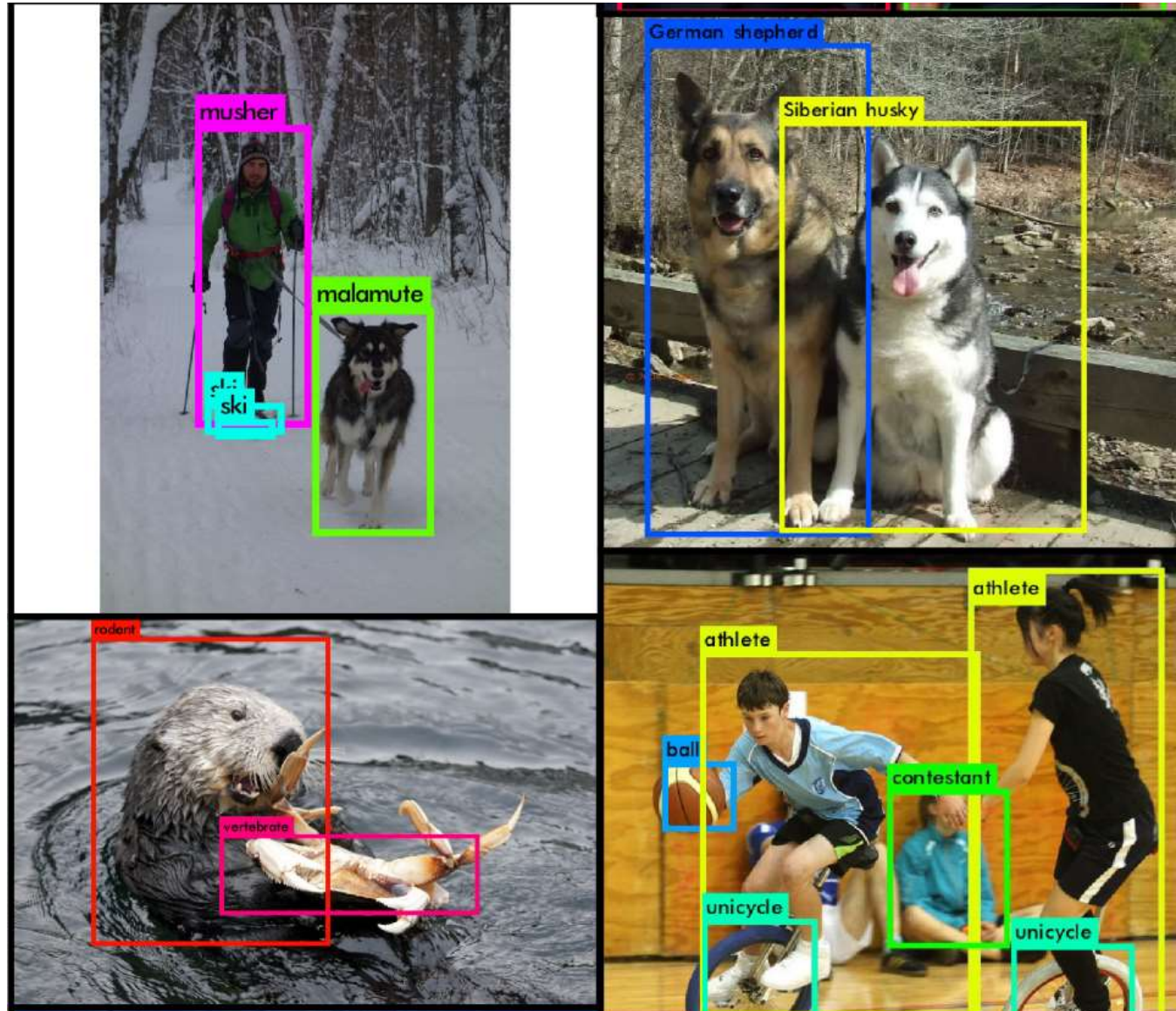
La version initiale de YOLO a été annoncée en 2015 dans la publication «[Vous ne regardez qu'une seule fois : détection d'objets unifiée en temps réel](#)» de Joseph Redmon, Santosh Divvala, Ross Girshick et Ali Farhadi.



Le YOLOv1 traitait les photos à 45 images par seconde en temps réel,

## 2. YOLOv2

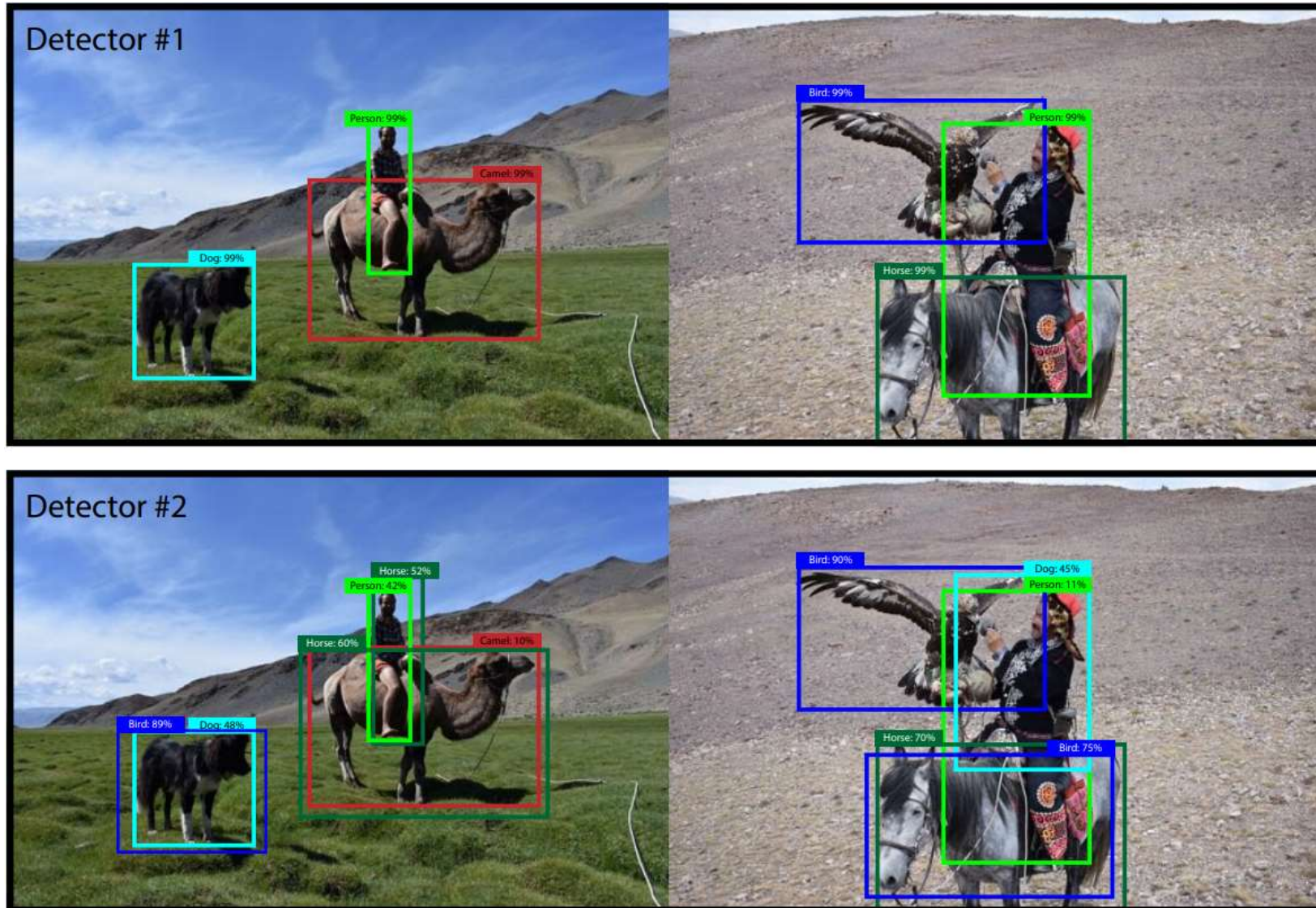
Un an plus tard, en 2016, Joseph Redmon et Ali Farhadi ont publié YOLOv2 (également connu sous le nom de YOLO9000) dans le journal "[YOLO9000 : meilleur, plus rapide, plus puissant.](#) »





### 3. YOLOv3

Dans le journal "[YOLOv3 : une amélioration progressive](#)", Joseph Redmon et Ali Farhadi ont publié une nouvelle version de l'algorithme en 2018. Il a été construit sur l'architecture Darknet-53. Des classificateurs logistiques indépendants ont remplacé le mécanisme d'activation softmax dans YOLOv3.



## Principe

- ✓ L'intelligence artificielle est un domaine de l'informatique dans lequel le programme apprend par lui-même à effectuer certaines tâches. Notamment de la reconnaissance visuelle.
- ✓ Nous allons utiliser un réseau de neurones entraîné pour reconnaître des formes particulières.
- ✓ Il faut de nombreuses données pour pouvoir entraîner correctement un réseau de neurone.
- ✓ Il a été démontré que l'apprentissage était plus rapide sur un réseau de neurones entraîné pour autre chose. Par exemple, un réseau de neurones entraîné pour reconnaître les chiens s'entraînera plus facilement à reconnaître les chats.

## 1- Configuration de Python

installer les librairies nécessaires imutils, OpenCV, ultralytics

```
python -m pip install imutils opencv-python ultralytics
```



## 2- Récupérer un modèle pré-entraîné :

Il est possible de récupérer un modèle pré-entraîné à partir du script python

```
# load the pre-trained YOLOv8n model  
model = YOLO("yolov8n.pt")
```

## 3- Création d'un flux vidéo

Création d'un flux vidéo (vs) à l'aide de la librairie imutils qui va récupérer les images de la caméra.

```
vs = VideoStream(src=0, resolution=(1600, 1200)).start()
```

## 4- Script Python pour la reconnaissance d'Objet

## 5- Sources d'image pour la détection d'objet

- La webcam de votre ordinateur

```
1. vs = VideoStream(src=0, resolution=(1600, 1200)).start()  
2. while True:  
3.     frame = vs.read()
```

Le stream vidéo doit être arrêté à la fin du script avec `vs.stop()`

- Une caméra IP

```
1. vc = cv2.VideoCapture('rtsp://user:password@ipaddress:rtspPort')  
2. while True:  
3.     ret, frame=vc.read() #from ip cam
```

Veillez à arrêter la capture vidéo à la fin du script avec `vc.release()`

## 5- Sources d'image pour la détection d'objet

- La Picam du Raspberry Pi

```
1. vs = VideoStream(usePiCamera=True, resolution=(1600, 1200)).start()  
2. while True:  
3.     frame = vs.read()
```

Pensez à arrêter le stream à la fin du script avec `vs.stop()`

- Un fichier vidéo

```
1. vc = cv2.VideoCapture('./datasets/Splash - 23011.mp4') #from video  
2. while True:  
3.     ret, frame=vc.read() #from video
```

- Un fichier image

```
1. frame= cv2.imread('./datasets/two-boats.jpg')
```