

# Project Deep Learning YOLOv8

Delphine ALLANO, Marion DEFFEE,  
Romain TAMIC, T rence ROUSIC



# Sommaire :

- Contexte
- Présentation YOLOv8
- Travaux effectués
- Problèmes rencontrés
- Visionnage de la vidéo



# Contexte :

---

Dans le cadre de notre cours de Deep Learning nous avons dû faire un programme Python qui exploite la puissance de YOLOv8n-pose. Notre objectif était de détecter les membres d'un humain dans une vidéo, en mettant un focus particulier sur les mains. En utilisant cette détection, nous avons réussi à superposer un tableau virtuel entre les mains de la personne, qui se déplace synchroniquement avec ses mouvements.



# Présentation Yolov8

- Huitième version de l'algorithme "You Look Only Once", système de détection d'objets en temps réel qui peut identifier des objets dans des images ou des flux vidéo avec une grande vitesse de précision
- Adapté aux applications nécessitant une analyse d'image en temps réel tels que la surveillance vidéo ou la reconnaissance de scènes en conduite autonome.
- La version 8 de YOLO améliore la précision et la rapidité de détection des objets.

# Fonctionnement YOLOv8

## Etape 1 : Partitionnement d'une image

L'image est divisée en une grille. Chaque cellule de cette grille est responsable de la détection des objets dont le centre tombe à l'intérieur de la cellule.

## Etape 2 : Extraction de caractéristiques

YOLOv8 utilise un réseau de neurones convolutif (CNN) pour extraire les caractéristiques de l'image à travers plusieurs couches. Ce CNN est optimisé pour détecter des caractéristiques complexes à différents niveaux d'abstraction.

## Etape 3 : Prédiction des boîtes englobantes et des classes

Pour chaque cellule de la grille, le modèle prédit plusieurs boîtes englobantes et leurs confiance associé, indiquant la probabilité qu'une boîte contienne un objet. En parallèle, il prédit également les probabilités de classe pour les objets détectés.

## Etape 4 : Traitement post-prédiction

Parmi les multiples boîtes prédites pour un même objet, celles avec une faible confiance sont éliminées,





# Entrainement de YOLOv8

- Lancée en 2009, ImageNet est une vaste base de données d'images conçue pour être utilisée dans des projets de recherche en vision par ordinateur.
- Plus de 14 millions d'images annotées à la main et réparties en plus de 20 000 catégories
- Chaque image est annotée avec un ou plusieurs labels indiquant les objets présents dans l'image



# Le modèle YOLO que nous avons utilisé

---

- YOLOv8n-pose est le modèle YOLO que nous avons utilisé. C'est une variante qui inclut la capacité de reconnaître des poses humaines. Cela rend donc ce modèle utile pour les applications nécessitant la détection de la position et du mouvement du corps humain.
- Autres modèles : détection, segmentation, classification...



# Travaux effectués :

Détection des keypoints sur le haut du corps (épaules, coudes, mains)



```
# Récupérer les coordonnées des épaules, coudes et mains
left_shoulder = xy[6] # Indice 6 correspondant à l'épaule gauche
left_elbow = xy[8] # Indice 8 correspondant au coude gauche
left_hand = xy[10] # Indice 10 correspondant à la main gauche
right_shoulder = xy[5] # Indice 5 correspondant à l'épaule droite
right_elbow = xy[7] # Indice 7 correspondant au coude droit
right_hand = xy[9] # Indice 9 correspondant à la main droite
left_eye = xy[1] # Indice 1 correspondant à l'œil gauche
right_eye = xy[2] # Indice 2 correspondant à l'œil droit

shoulder_to_elbow_left = draw_line_between_keypoints(frame, left_shoulder, left_elbow)
elbow_to_hand_left = draw_line_between_keypoints(frame, left_elbow, left_hand)
shoulder_to_elbow_right = draw_line_between_keypoints(frame, right_shoulder, right_elbow)
elbow_to_hand_right = draw_line_between_keypoints(frame, right_elbow, right_hand)
eye_to_eye = draw_line_between_keypoints(frame, left_eye, right_eye)
```





# Travaux effectués :

---

- On insère une image dans la vidéo en l'insérant frame par frame
- La fonction pour insérer entre les keypoints 9 et 10 :

```
def draw_line_between_keypoints(frame, keypoint1, keypoint2):  
    """ Dessine une ligne entre deux keypoints sur le frame. """  
    keypoint1 = tensor_to_tuple(keypoint1)  
    keypoint2 = tensor_to_tuple(keypoint2)  
    cv2.line(frame, keypoint1, keypoint2, color=(0, 255, 0), thickness=2)
```

# Travaux effectués :

---

- On peut insérer un tableau maintenant !





# Travaux effectués :

---

- On peut faire une boucle pour afficher le tableau pour chaque personne présente sur la vidéo

# Travaux effectués :

---

- En regardant les keypoints 1 et 2 (œil gauche et droit) on peut définir les coordonnées "plafond" pour faire apparaître le tableau et le faire disparaître

Cela peut se faire avec deux variables d'état :

- Show : permet d'afficher le tableau
- Toggled\_allowed : permet de garder en mémoire si les mains sont passées au-dessus des yeux





# Problèmes rencontrés :

- Méconnaissance de YOLOv8 -> période de documentation et de tests
- Ecriture sur une vidéo frame par frame
- Détection des keypoints
- Gérer l'apparition du tableau avec les coordonnées des keypoints
- Mettre un tableau par personne

**Avez-vous des questions ?**

