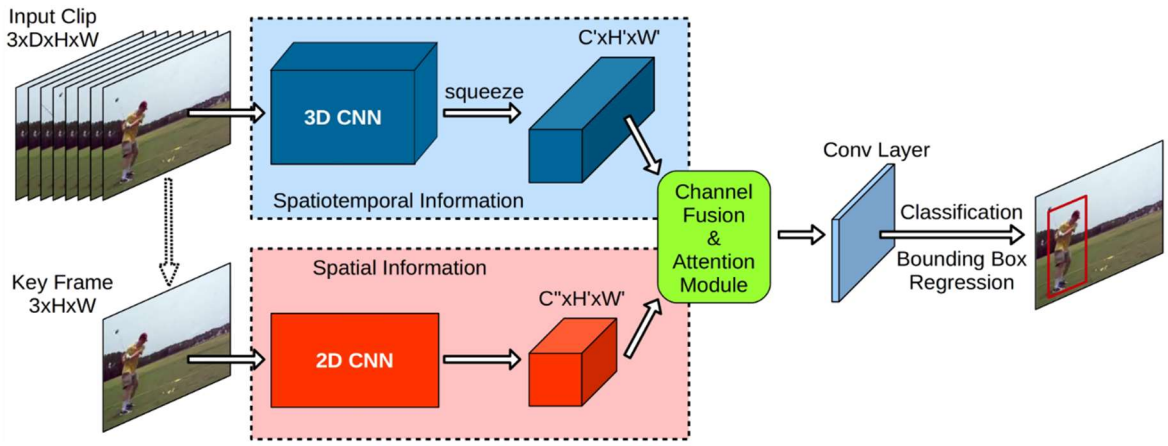
You Only Watch Once

Détection d'actions

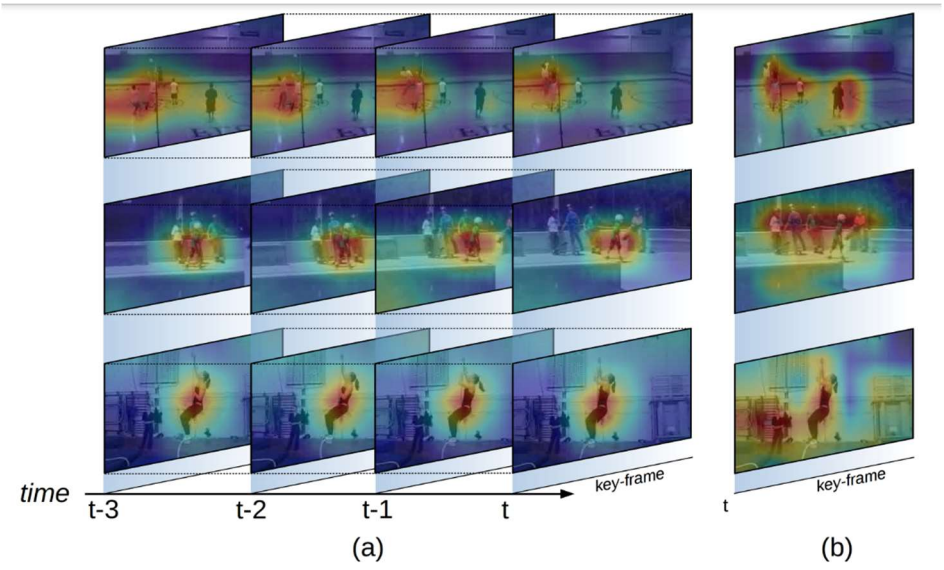
Fonctionnement :



backbone 2D: Permet de détecter la personne.

backbone 3D: Permet de détecter l'action.

attention : Aggrégation des caractéristiques provenant du backbone 3D avec celles provenant du backbone 2D pour suivre les personnes et leurs actions.



Modification du code pour l'adapter à votre machine :

dataset/ava.py : -ligne 24 (Chemin d'accès vers train.csv) & ligne 27 (Chemin d'accès vers val.csv)

- ligne 77 (Chemin d'accès vers les vidéos d'une seconde)

evaluator/ava_evaluator.py :-ligne 40 (Chemin d'accès vers val.csv)

train_finetuning_ava.py :- ligne196 (Chemin d'accès vers les poids du modèle YOWO Nano entraîné sur Ava)

-ligne 144 (key WandB)

Classes :

1 : "SL"

2 : "NR"

Rôle des classes :

class dataset--->ava.py(classe Dataset Pytorch)

--->transform.py

evaluator--->ava_eval_helper.py (Classe complémentaire pour ava_evaluator

--->ava_evaluator.py (Étapes de validation)

--->cal_frame_mAP.py (calcul de métrique mAP_frame)

--->cal_video_mAP.py (calcul de métrique mAP_video)

--->utils.py

Fonction de perte (Loss) :

Loss détection

- L1,smooth(x, y) = 0.5(x − y) ^2 if|x − y| < 1

  |x − y| − 0.5 otherwise

  ➔ perte L1 qui est **moins sensible aux valeurs aberrantes et aux gradients explosifs**. Elle est utilisée pour calculer la perte associée à **la localisation des objets** dans le modèle.

- *LMSE*(*x*,*y*)=(*x*−*y*)^2

  ➔ calculer l'erreur entre **les scores de confiance** prédits par le modèle et les scores de confiance réels des données d'entraînement.

➔LD=Lx+Ly+Lw+Lh+Lconf = Llocalisation +Lconf

Loss classification :

- *Lfocal*(*x*,*y*)=*y*(1−*x*)^*γ* log(*x*)+(1−*y*)*x*^*γ* log(1−*x*)
  - x : Les prédictions de probabilité de classe générées par le modèle.
  - y : Les étiquettes de classe réelles (0 ou 1) correspondant aux données d'entraînement.
  - γ : Le facteur de modulation qui contrôle la pondération des exemples en fonction de leur confiance.

➔Elle est conçue pour traiter le problème du déséquilibre de classe dans les ensembles de données

➔Modifie la perte en fonction de la confiance de prédiction

Entraînement d'un modèle ajusté avec la base de données Ava :

Cd

pip install -r requirements.txt

!python train_finetuning_ava.py --cuda -d ava_v2.2 -v yowo_nano --num_workers 4 --eval_epoch 1 --eval

Distributed train :

'-dist', '--distributed' : action='store_true', default=False, help='distributed training'

'--dist_url' : default='env://', help='url used to set up distributed training'

'--world_size' : default=2, type=int,help='number of distributed processes'

'--sybn' : action='store_true', default=False, help='use sybn.')