Rapport du projet vidéo

YOLO: Unified, Real-Time Object Detection & Seq-NMS for Video object Detection

Yunyun SUN, Yutong YAN, Sixiang XU, Heng ZNANG

Problématique 2

Cahier des charges 3
- **Étapes principles** 3
- **Plateforme du système** 3
- **Bibliothèques utilisées** 3
- **Code source** 3

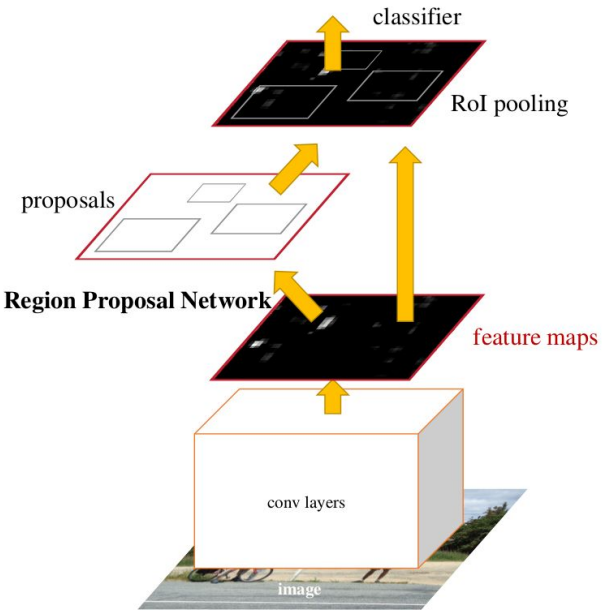
La réalisation 4
- **Architecture du systeme** 4
- **You Only Look Once: Unified, Real-Time Object Detection** 5
  - **Idée de base YOLO** 5
  - **Les principales caractéristiques de YOLO** 5
  - **Processus général** 5
  - **Structure du réseau** 6
  - **Training** 7
  - **Test** 8
- **Seq-NMS for Video object Detection** 9
  - **Introduction** 9
  - **Les approches Seq-NMS** 9
  - **Quelques exemples** 12

Bilan 13
- **Atouts et Limites** 13
  - **Atouts** 13
  - **Limites** 14

Démo 15
- **Démo 1: Vidéo avec beaucoup de mouvements** 15
- **Démo 2: Vidéo de surveillance dans un métro** 15

Problématique

Les performances de l'état de l'art en détection d'objets sur une image ont été considérablement améliorées au cours des deux dernières années. D'une part, l'évolution des structures du CNN(AlexNet[1], GoogLeNet[2], ResNet[3], NasNet[4], etc) a amélioré la capacité d'extraction des features sur une image, de l'autre part, le développement sur le framework de détection d'objet permet de mieux utiliser ces features(meilleur précision ou plus rapide).



Une exemple de Faster R-CNN framework pour la détection d'image

La différence entre la détection d'image et la détection de la vidéo est l'existence des informations temporelles et contextuelles[5]. Ce n'est pas convenable de décomposer une vidéo à des frames et appliquer ces frameworks de détection d'image l'un après l'autre. Par contre, il faut spécialement désigner un approches pour combiner les features d'image et ses informations temporelles et contextuelles.

Avec ces informations temporelles et contextuelles, nous pouvons: soit accélérer la framwork, soit améliorer la précision de framwork.

[1] ImageNet Classification with Deep Convolutional Neural Networks
[2] Going Deeper with Convolutions
[3] Deep Residual Learning for Image Recognition
[4] Learning Transferable Architectures for Scalable Image Recognition
[5] T-CNN: Tubelets with Convolutional Neural Networks for Object Detection from Videos

Cahier des charges

Notre projet vise à réaliser un système de détection d'objet dans la vidéo.

Concrètement, nous avons conservé la framework de détection d'image(YOLO[6][7]) autant que possible et ajouté un post-traitement(Seq-nms[8]) pour améliorer la robustesse de mouvement d'objet.

Étapes principles

1. **Extraction d'images à partir d'une vidéo;**
2. **Effectuer la détection sur chaque frame extrait en utilisant l'approche YOLO;**
3. **Filtrer les résultats de l'étape précédente avec l'algorithme seq-nms.**
4. **Reconstruire la vidéo à partir des résultats de détection filtrées.**

Plateforme du système

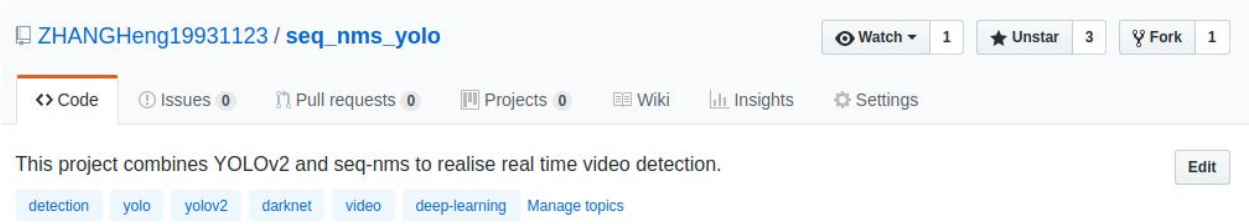
- **Ubuntu 16.04 64bit**
- **Intel core i7-4700MQ**
- **Nvidia GT 750M(2GB GDDR6)**

Bibliothèques utilisées

- **Tensorflow object detection api[9]**
- **Darknet[10]**
- **Seq-NMS[11]**

Code source

Le code source est publié sur [github](#).



[6] You Only Look Once: Unified, Real-Time Object Detection
[7] YOLO9000: Better, Faster, Stronger
[8] Seq-NMS for Video Object Detection
[9] models
[10] darknet
[11] Seq-NMS

La réalisation

Architecture du systeme

You Only Look Once: Unified, Real-Time Object Detection

Pour la réalisation, on utilise une approche de reseau neurone convolutif YOLO.

YOLO est un réseau de neurones intelligent pour la détection d'objets en temps réel.

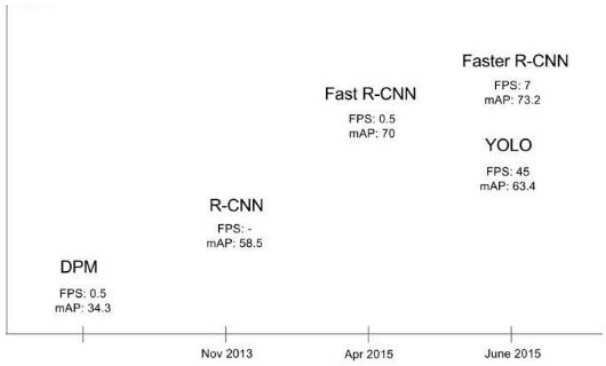
On dit que c'est en temps réel: parce que il est très rapide, jusqu'a 45 fps, mais effectivement le traitement est encore frame par frame, donc pour la vraie vidéo détection, nous avons appliqué une autre technique sur cette base: Seq-nms.
YOLO adopte une approche complètement différente de ce que nous avons appris auparavant. YOLO ne regarde qu'une seule fois l'image (d'où son nom: You Only Look Once) mais d'une manière intelligente.

Idée de base YOLO

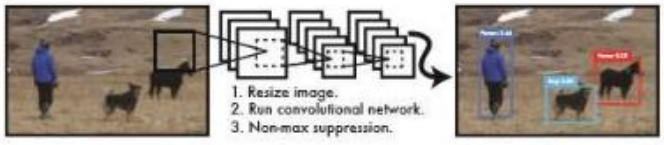
De R-CNN à Faster R-CNN l'idée qui a toujours été retenue est la proposal + la classification (proposal fournit des informations de localisation et la classification fournit des informations sur les catégories). La précision est très élevée mais la vitesse est insuffisante. YOLO offre un moyen plus direct de revenir directement la position du bounding box à la couche de la sortie et la catégorie à laquelle appartient le bounding box (l'image entière sert d'entrée au réseau, transformant le problème de détection d'objet en question de régression)

Les principales caractéristiques de YOLO

- **Rapide, capable de détection en temps réel. Jusqu'à 45 fps sur le GPU de Titan X**
- **End to end training.**
- **Utilisez l'image complète en tant qu'information contextuelle, avec relativement peu d'erreur d'arrière-plan (le background est incorrectement reconnu en tant qu'objets).**
- **Capacité de généralisation (la généralisation d'images naturelles à d'autres domaines).**



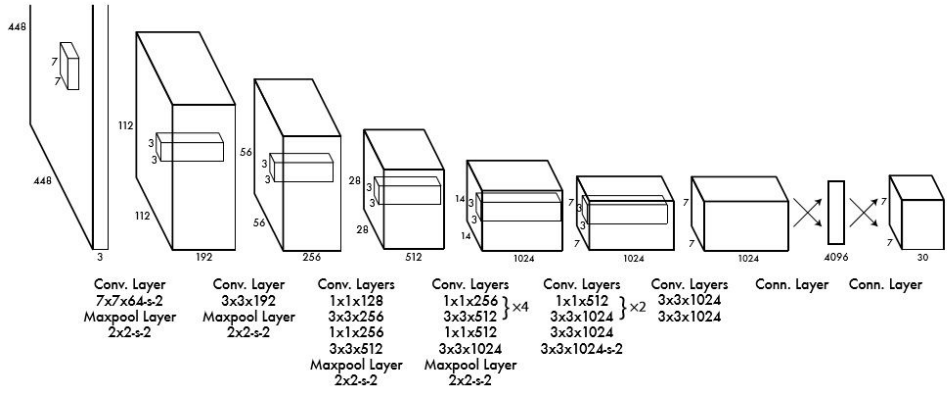
Processus général



1. **Redimensionner l'image d'entrée à 448*448, segmenter l'image a cellules de 7*7.**

2. **Extraction des caractéristiques et de prédiction par CNN: La partie convolution est responsable de l'extraction des caractéristiques. La partie de fully-connected est responsable de la prédiction: a) les coordonnées (x_{center}, y_{center}, w, h) de 98 bounding-box (7 * 7 * 2) et la confiance de l'objet. b) La probabilité de 20 objets appartenant à 49 cellules.**
3. **Filtrage du bounding box (via nms).**

Structure du réseau



La structure du réseau s'appuie sur GoogleNet[12]. 24 couches convolutifs, 2 couches de fully connected. Le module d'inception de Googlenet est remplacé par 1×1 reduction layers suivi de 3×3 convolutional layers.

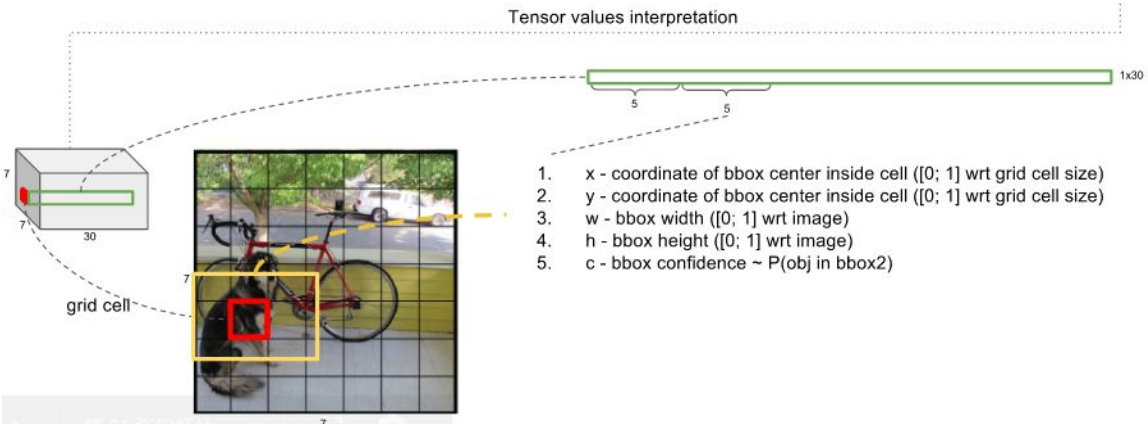
[12] **C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842, 2014.**

Layer	kernel	stride	output shape
Input			(416, 416, 3)
Convolution	3×3	1	(416, 416, 16)
MaxPooling	2×2	2	(208, 208, 16)
Convolution	3×3	1	(208, 208, 32)
MaxPooling	2×2	2	(104, 104, 32)
Convolution	3×3	1	(104, 104, 64)
MaxPooling	2×2	2	(52, 52, 64)
Convolution	3×3	1	(52, 52, 128)
MaxPooling	2×2	2	(26, 26, 128)
Convolution	3×3	1	(26, 26, 256)
MaxPooling	2×2	2	(13, 13, 256)
Convolution	3×3	1	(13, 13, 512)
MaxPooling	2×2	1	(13, 13, 512)
Convolution	3×3	1	(13, 13, 1024)
Convolution	3×3	1	(13, 13, 1024)
Convolution	1×1	1	(13, 13, 125)

L'architecture de YOLO:
125 canaux pour chaque cellule:
- **Les données pour les bounding box**
- **Les prédictions de classe**
- **5 bounding boxes * 25 éléments de données (x et y, largeur, hauteur du rectangle de la boite;le score de confiance; la distribution de probabilité sur les classes)**

Training

1. **Pré-apprentissage du réseau de classification: Pré-apprentissage d'un réseau de classification sur ImageNet 1000-class competition dataset. Ce réseau est les premiers 20 réseaux convolutifs dans la figure 3 + average-pooling layer + fully connected layer (À ce moment, l'entrée du réseau est 224 * 224).**
2. **Apprentissage du réseau de détection:**



- **L'image est divisée en cellules de 7 x 7. Sur lequel se trouve le centre d'un objet, cette grille est responsable de la prédiction de cet objet.**
- **La grille est responsable à l'information de classification, le bounding-box est principalement responsable de l'information de coordonnées. Comme suit:**

Chaque grille (cellule de dimension 1*1*30) prédit les coordonnées (x_{center},y_{center},w,h) des deux bounding-box (Boîte de ligne solide jaune dans la figure). Parmi eux, x_{center} et y_{center} sont normalisés à 0-1 par rapport à la grille correspondante, w et h sont normalisés à 0-1 par rapport au largeur et hauteur dans l'image. En plus de retourner a son propre position,

chaque bounding box doit aussi retourner une valeur de confidence, qui représente la confiance de la contenance d'objet dans le bounding box predit, et la précision de ce box:

$$\text{confidence} = Pr(Object) * IOU_{pred}^{truth}$$

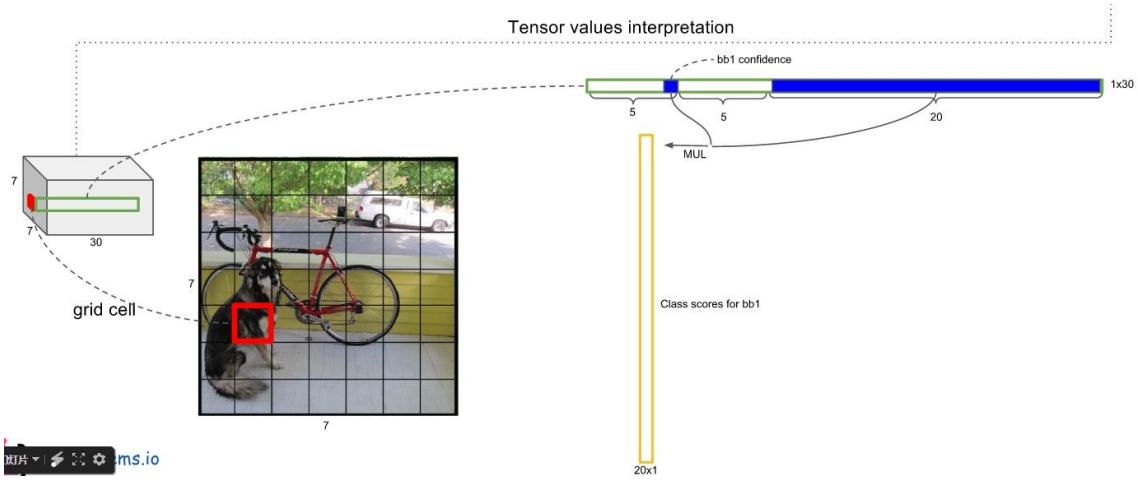
Si le bbox de la vérité terrain tombe dans un cellule de grille, le premier prendre 1, sinon prendre 0. Le deuxième est l'IOU entre le bbox predit et la vérité terrain. Chaque bounding box a cinq valeurs à prédire, et deux ont un total de dix valeurs, correspondant aux 10 premières caractéristiques dans 1 * 1 * 30.
Chaque grille prédit également des informations sur la classe. Il y a 20 classes dans l'article. Donc chaque cellule dans les 7*7 grilles doit predire 2 bbox et 20 classes. L'output est donc 7*7*(5*2+20). Note que l'information de classe est pour chaque grille, l'information de confiance est pour chaque bbox).

Test

Lors du test, l'information de classe prédit de chaque grille multiplie par l'information de confiance prédit de chaque bbox, on obtient le class-specific confidence score de chaque bbox.

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

Le premier élément sur le côté gauche de l'équation est l'information de classe pour chaque prédiction de grille, et le second élément est la confiance pour chaque prédiction de bbox. Ce produit encode la probabilité que la bbox prédite appartienne à une classe et aussi les informations de précision du box.



Seq-NMS for Video object Detection

Introduction

Les problématiques de la recognition des objets singulières dans les vidéos nous posent des problèmes très souvent. Les soucis les plus communs sont que dans certains frames les détections des objets sont pertinentes et faciles néanmoins les mêmes objets sont mal détectés dans autres frames dans le même morceau de la vidéo. Et les raisons qui causent des problèmes se varient, par exemple : (1) Le changement d'échelle est drastique. (2) L'occlusion. (3) flou de motion.

L'état de l'art pour la détection des objets singulières d'image peut se diviser vers trois phases distinctes : (1) génération de proposition de région. (2) classification des objets. (3) post-traitement. Pendant la phase de génération de proposition de région, un ensemble des candidatures des régions est généré selon leurs possibilités de contenir certains objets.

Tandis que l'état de l'art actuel, Faster R-CNN, apprends à générer des propositions avec des réseaux neurones. Les régions candidatures sont distribuées avec des notes de classification pendant la phase de la classification des objets. Et les détections redondantes sont filtrées par la suite pendant la phase de post-traitement.

Quand même efficace, les méthodes sont naïves parce qu'elle néglige totalement la dimension temporelle. Donc, dans la méthode seq-NMS, on combine des informations temporelles dans la phase de post-traitement visant à affiner des détections dans chaque frame indépendamment.

Avec une séquence d'une vidéo donné qui a des propositions des régions et leurs notes correspondantes de la classification, notre méthode va associer des bounding boxes dans les frames adjacents selon un critère de superposition simple. Et elle va sélectionner des boxes pour maximiser des notes de séquence. Les boxes choisis vont nous permettent de supprimer des autres boxes superposés respectivement dans leurs frames correspondants. Après, les boxes auraient des nouvelles notes visant à augmenter faibles détections.

Les approches Seq-NMS

La plupart des méthodes de détection d'objets sont conçues pour les performances de détection dans les frames indépendants. Cependant, puisque nous sommes concernés par la détection des objets dans vidéo, c'est moins logique à ignorer entièrement la composante temporelle.

En fait, on notice un problème très commun avec Faster R-CNN, c'est que la non-maximum suppression (NMS) va souvent choisir mal les bounding box après la classification des objets. Elle a une tendance à choisir des boxes trop grandes, qui aura une petite intersection-over-union (IoU) avec la boîte de vérité terrain à cause de l'union de régions

grande. Et les boxes grands normalement ont des notes élevées des objets, venant des riches informations qui sont prêtes à extraire.

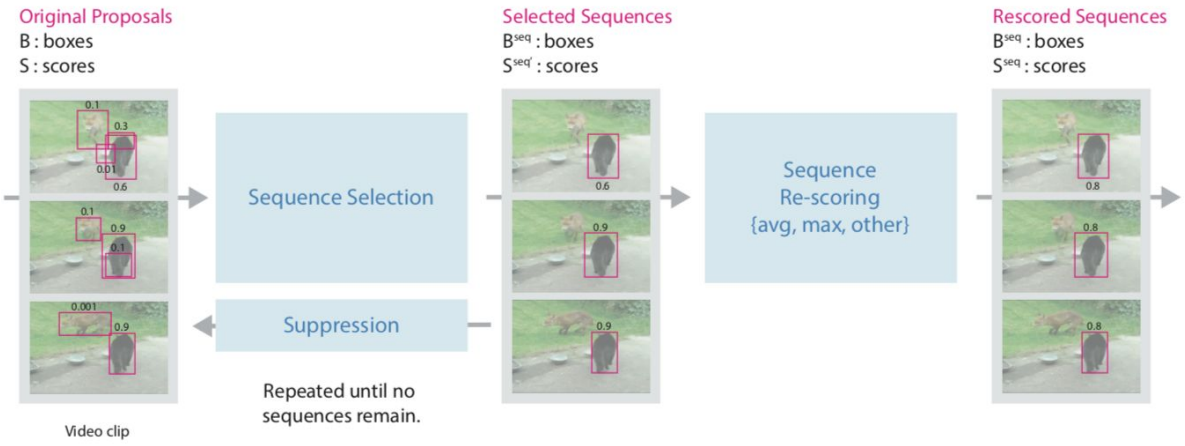
Visant à résoudre ce problème, on utilise les informations temporelles pour renouveler les notes des boîtes. Nous ainsi utilisons une méthode de re-classer les bounding boxes dans une vidéo séquence qui s'appelle Seq-NMS.

Seq-NMS a trois étapes principales :

1. **Sélection de séquence**
2. **Renouveler des notes de séquence**
3. **Suppression**

Nous répétons ces trois étapes jusqu'à la fin.

Illustration de Seq-NMS :



Seq-NMS prend toutes les boxes B des propositions des objets comme les entrées et noter S pour la vidéo entière V. (Contrairement, NMS prends des propositions d'une seule image .)

C'est un procédé itératif, il y a trois étapes dans chaque itération :

1 Sélection de séquence

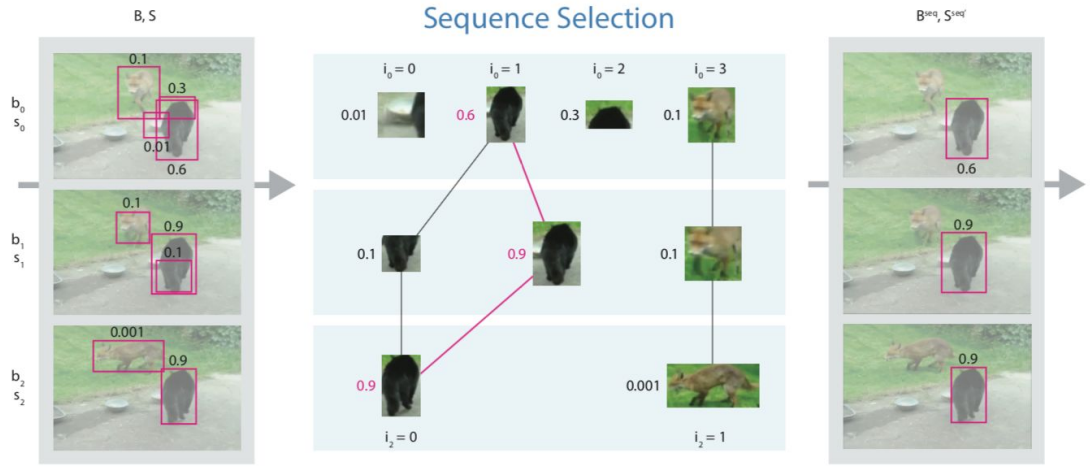
Il sélecte une séquence des boxes dont les notes sont plus élevées, B^{seq} **.**

Pour chaque pair de frames voisins dans V, un bounding box dans le premier frame peut avoir un lien avec le bounding box dans le frame suivant sii leur IoU passe un certain seuil. On voudrait trouver un lien possible dans chaque pair de frames voisins parmi la vidéo entière. C'est-à-dire qu'on voudrait trouver la séquence de box qui maximise la somme des notes de l'objet, avec la contrainte que tous les boxes adjacents doivent être liés.

$$
\begin{aligned}
i' \quad &= \quad \underset{i_{t_s},\ldots,i_{t_e}}{\operatorname{argmax}} \sum_{t=t_s}^{t_e} s_t[i_t] \\
&s.t. \quad 0 \le t_s \le t_e < T \\
&s.t. \quad IoU(b_t[i_t],\ b_{t+1}[i_{t+1}]) > 0.5,\ \forall t \in [t_s, t_e)
\end{aligned}
$$

on peut utiliser un algorithme simple dynamique de programmation qui peut maintenir la note maximale jusqu'à maintenant dans chaque box. L'optimisation retourne un ensemble des indices i' qui sont utilisées pour extraire une séquence de box B^{seq} = { $b_{t_s}[\,i_{t_s}\,]$, ... , $b_{t_e}[\,i_{t_e}\,]$ } et leurs notes $S^{seq'}$ = { $S_{t_s}[\,i_{t_s}\,]$, ... , $S_{t_e}[\,i_{t_e}\,]$ }.

Illustration de sélection de séquence :



on construit un graphe où les boxes de frames adjacents sont liés sii leur IoU > 0.5 .

Une séquence est un ensemble de boxes qui sont liés dans la vidéo. On donc sélecte la séquence avec la note de séquence la plus élevée qui est affichée dans l'équation au-dessous. On obtiendra B^{seq} et $S^{seq'}$ qui est un ensemble d'un box par frame au plus et les notes associées. Après la sélection de séquence, pour chaque box dans B^{seq} , on supprime tous les boxes dans le même frame dont IoU > 0.3.

2 Renouveler des notes de séquence

Il prend toutes les notes dans la séquence $S^{seq'}$ et les applique une fonction visant à obtenir une nouvelle note pour chaque frame dans la séquence S^{seq} .

Après la séquence est sélectionnée, les notes dedans sont améliorées. On applique une fonction F à la séquence de notes pour produire S^{seq} = F(S^{seq}) .

3 Suppression

Pour chaque box dans la séquence B^{seq} , on supprime tous les boxes dans la même frame qui ont les superpositions suffisantes. Si un bounding box dans frame t , t_s < t < t_e ,a un IoU avec b_t supérieur à un certain seuil, il sera supprimé de l'ensemble de boxes candidature .

Quelques exemples

Il y a plusieurs travails dans la détection de l'objets dans vidéo qui sont dans le cadre de tracking des objets multiples. Il y a un sous-classe populaire des techniques qui est un modèle faisant 'tracking-par-détection'. Où un algorithme de détection est appliqué à chaque vidéo frame et la détection sera associée à travers les frames pour générer la trajectoire de chaque objet.

L'exemple de vidéo où Seq-NMS améliore des performances.

Illustration :

Les boxes représentent une séquence sélectionnée par Seq-NMS. Les vidéo clips sont sous-échantillonnées pour fournir des exemples de boxes de notes élevées et de notes faibles.

Dans les clips a, b, et e, l'objet devient plus en plus occlus à cause de l'existence de frame, nous amenant vers des notes plus faibles. Cependant, dans les clips c et d, l'objet d'intérêt a une note de classification plus faible parce qu'il est petit ou flou respectivement. Dans tous les cas, Seq-NMS' qui renouvelant les notes significativement amélioré des détections faibles en utilisant les détections plus pertinentes des frames adjacents.

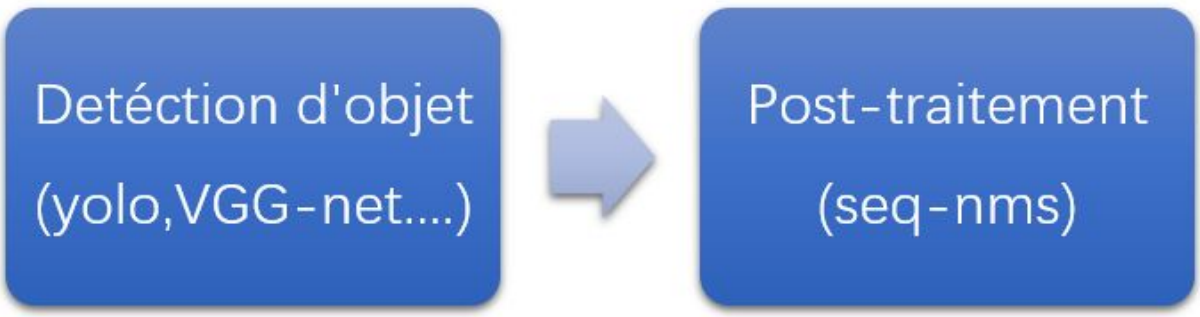
Bilan

Atouts et Limites

Atouts

Implémentation

En tant qu'un algorithme de détection d'objet dans une vidéo, son implémentation est plus facile. Sa fonction est bien séparée de la part de détection d'objet sur les frames et elle juste améliore les résultats (les bounding boxs et les scores) de détection en utilisant les informations temporelles entre les frames. Autrement dit, on peut mettre en œuvre cet algorithme à l'étape de post-traitement et avant de lui, n'importe quelle méthode de la détection d'objet sur l'image qui génére les régions d'intérêt s'adapte. Dans notre projet, la méthode de détection d'objet sur l'image est « Yolo », par contre dans la thèse que l'on consulte, c'est VGG-Net. Au fur et à mesure que l'évaluation d'algorithme de détection d'objet, sans aucun doute, la méthode seq-nms devient de plus en plus performante.



Suivi pertinent

Dans une vidéo, un objet que l'on souhaite suivre peut-être se déplace rapidement. À la cause de la qualité de l'imagerie de caméra, il y a des flous autour de l'objet qui dégradent l'effet de détection et font baisser le score de confiance sur une grande échelle. Si on utilise d'autres méthodes conventionnelles de post-traitement, par exemple nms, les résultats de détection dont le score est faible vont être supprimés et le suivi ne marchera pas.

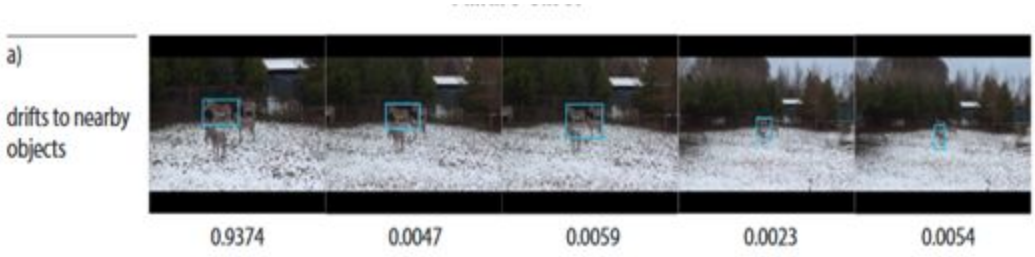
Contrairement, la méthode seq-mesure va conserver les résultats dont leurs scores sont fables puisqu'elle réfère aussi à la division temporelle et elle ne compte que le score total dans une séquence au travers des frames. Egalement, selon la même raison, seq-nms résout d'autres problèmes (tels que : occlusion, taille petite) qui perturbent le suivi.

Limites

Non en temps réel

Malheureusement, le traitement du vidéo de seq-nms n'est pas en temps réel. Il faut tout d'abord prendre une vidéo et puis appliquer seq-nms sur telle vidéo que le nombre de frame ne pourra plus augmenter.

Suivi dégradé en certain cas



Comme l'illustration au-dessus présente, si des objets en même classe se concentre dans une espace petite et ils nous rendent combles, c'est très difficile pour seq-nms de suivre continuellement un objet particulier. En conséquence, les autres objets proches de la cible peuvent être éventuellement suivis. Intuitivement, un seuil de IOU plus élevé nous permet de sélectionner un bounding box qui est plus corrélé spatialement avec le bounding box de frame adjacent. Donc l'augmentation de la valeur de seuil de IOU est une solution potentielle. Cependant, Il exige le changement très faible de la position de l'objet à suivre entre les frames adjacents. En plus, cette solution marche mal si l'objet se déplace très rapidement ou l'objet et d'autres objets similaires se superposent devant la caméra.

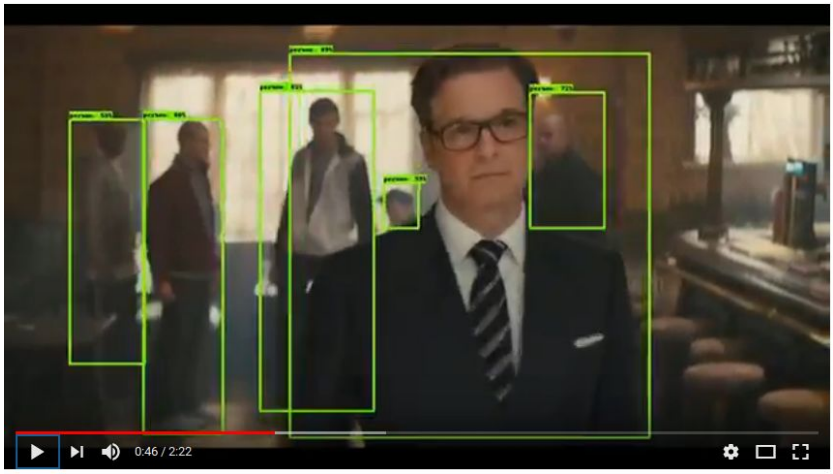
Dans un autre exemple de l'échec illustré au-dessus, le détecteur a perdu la cible dans les trois premières images. Les bounding box ne sont pas satisfaits du tout et les scores de confiance sont toujours inférieur à 0.5. En observant les deux dernières images dont les bounding boxs ont du score élevé, on s'aperçoit que peut-être la voiture rouge est la cible.

En fait, seq-nms n'adopte aucune méthode pour pénaliser les détections « fausses-positives » et elle entraîne une situation déplaisante où des détections fausses-positive des frames adjacents se relient et deviennent une séquence sélectionnée dans laquelle il existe aussi des détections « vrais-positives ».

Démo

Cliquez les images pour regarder les vidéos de démonstration sur Youtube.

Démo 1: Vidéo avec beaucoup de mouvements



Démo 2: Vidéo de surveillance dans un métro