Détection et suivi de colis sur convoyeur

Amédée Holy
amedee.holy@student.ecp.fr

Abstract

Le suivi automatique des colis sur tapis roulant est jugé comme un outil indispensable pour les entreprises du secteur et les permet aux entreprises avec un retard dans la distribution d'être compétitif sur le marché des colis. L'objectif de ce projet qui est également la solution au besoin d'un client du secteur de distribution de colis est donc de chercher et de développer une librairie de suivi d'objets. Le programme de suivi devra détecter et suivre les colis dans différentes situations. Le but du projet est de trouver un algorithme de tracking d'objet sur un flux vidéo. En pratique, on cherche à obtenir un algorithme qui respecte les conditions suivantes :

• Flux vidéo en temps réel en entrée • Résultat en temps réel(>2Hz)
• Permet le suivi de plusieurs objets

1. Introduction

Aujourd'hui, dans la vision par ordinateur la classification d'images et la détection multiple d'objets sur des images ou un flux vidéo est une tâche très bien maîtrisée, les algorithmes basés sur des réseaux de neurones profonds (Deep Learning) ayant atteints des précisions inégalées dans les dernières années (GoogLeNet, ResNet, . . .).

La tâche qui en découle, le suivi d'objet, autant individuel que multiple demeure toutefois un domaine où les résultats, bien que bons, disposent encore d'une marge de progression conséquente. Les approches disponibles sont nombreuses et diverses.

Les difficultés du suivi d'objet sont multiples :

- Occlusion : il est possible que certains objets soient masqués à certains intervalles de temps et doivent ensuite être ré-associés correctement par la suite
- Déformation : les objets peuvent avoir une forme qui change au cours du temps
- Vitesse : les objets peuvent avoir des vitesses de déplacement variables et rapides rendant le suivi difficile
- Flux : le nombre d'objet à suivre simultanément sur une même image peut être très variable et très important

2. Méthodes de tracking

Le tracking by detection est aujourd'hui le paradigme le plus populaire. Il permet notamment une corrélation entre les différentes étapes du tracking.
En effet l'algorithme est découpé en étapes distinctes :

- Détection des objets voulus
- Prédiction de la position suivante des objets
- Association entre les détections à 2 instants successifs vis-à-vis des prédictions

Les réseaux de neurones quant à eux jouissent d'une popularité récente et sont parmi les méthodes les plus prometteuses. En effet, leur utilisation efficace en détection d'objet et classification d'image laisse penser à des applications dans le domaine du suivi d'objets.

Les méthodes à base de réseaux de neurones récurrents (RNN) dont le fonctionnement semble bien se prêter à ce type de problème.
Ils permettent entres-autres de résoudre les 2 tâches du tracking du tracking By Détection : la prédiction et l'association.

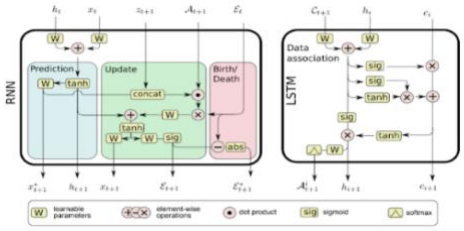


Figure - exemple de réseaux récurrents pour la prédiction (gauche) et l'association (droite)
Online Multi-Target Tracking Using Recurrent Neural Networks, A. Milan & al.

La suite de cette étude se basera principalement sur les deux paradigmes présentés ci-dessus.

3. Indicateur de performance

Dans le cadre du tracking d'objet sur flux vidéo, plusieurs indicateurs permettent d'évaluer la performance des modèles et donc de les comparer.

Indicateur	Explication
MOTA	Combinaison de 3 erreurs : $1 - \frac{\sum_t fn_t + fp_t + id\_sw_t}{\sum_t g_t}$ (gt : nb objets / fnt : nb faux négatifs / fpt : nb faux positifs / id_swt : nb id inversés)
MOTP	Mesure la concordance des bounding box : $\frac{\sum_{i,t} d_t^i}{\sum_t c_t}$ d_t^i : distance entre vérité terrain et prédiction pour objet I à frame t / c_t : nb association frame t
MT	Nombre d'objet trackés pendant plus de 80% de leur temps de vie
ML	Nombre d'objet trackés pendant moins de 20% de leur temps de vie
FP	Faux positifs
FN	Faux négatifs
IDS	Nombre de fois ou un objet se voit assigné un mauvais ID
Frag	Nombre de coupure de trajectoire
Hz	Fréquence en fps

Les paramètres FP, FN et MOTP sont plus particulièrement liés au détecteur utilisé. Les performances globales de l'algorithme de tracking dépendent aussi de la qualité et de l'exactitude de la détection. En ce sens, le paramètre MOTA prend en compte l'ensemble des erreurs de l'algorithme.

4. Etat de l'art

Voici certaines des méthodes jugées prometteuses après plusieurs recherches.

Deep Continuous Conditional Random Fields with Asymmetric Inter-object Constraints [1]

Utilise Deep Continuous Conditional Random Field (DCCRF). Le DCCRF reçoit en entrée l'image au temps t et au temps t-1 ainsi que les trajectoires des objets suivis jusqu'au temps t-1.
Il prédit ensuite la position au temps t des objets. Pour ceux il utilise 2 paramètres :
- des termes unaires pour modéliser les mouvements individuels des objets, obtenus via un CNN qui est entraîné pour estimer les déplacements de chaque objet entre t-1 et t.
- des termes asymétriques par paires pour les interactions entre les objets qui visent à gérer les occlusions et plus généralement les erreurs et le bruit.

L'association se fait à l'aide de l'algorithme hongrois, les poids étant définis par une mesure de similitudes réalisée à l'aide d'un CNN, entre les détections au temps t-1 et t.

Gestion des occlusions

Les occlusions sont gérées par un compteur : si un objet disparaît, sa position devient une position virtuelle (ie celle prédite). Si l'objet raccroche, la liaison est interpolée. Sinon au bout de m frames il est considéré comme ayant disparu.

Performances

	MOTA	MOTP	IDS	MT	ML	HZ
2DMOT15	33.6	70.9	866	10.4	37.6	online
2DMOT16	44.8	75.6	968	14.1	42.3	online

Deeply Learned Candidate Selection and Person Re-identification (MOTDT) [2]

La méthode se fait en 2 étapes, tout d'abord on réalise une sélection minutieuse de candidats pour limiter le volume computationnel. On réalise ensuite l'association.

En utilisant un R-CNN (region-based CNN), l'algorithme crée des cartes de score pour chaque région d'intérêt et obtient des feature caractéristiques de des objets
La position des objets au temps t est ensuite estimée à l'aide d'un filtre de Kalman.
Enfin, l'algorithme utilise un seuil IoU (Intersection over Union) sur les bounding box des positions prédites et des régions d'intérêt pour sélectionner les candidats.
Un score de similarité est calculé à l'aide d'un CNN basé sur la structure de GoogLeNet.
L'association des données se fait de façon hiérarchique basée sur ce score.

Gestion des occlusions

La gestion des occlusions se fait avec le filtre Kalman pour prédire la position d'objets cachés.

Performances

	MOTA	MOTP	IDS	MT	ML	Hz
MOT16	47.6	74.8	792	15.2	38.3	20.6
MOT17	50.9	76.6	2474	17.5	35.7	18.3

Motion Segmentation & Multiple Object Tracking by Correlation Co-Clustering [3]

L'algorithme utilise deux méthodes déjà existantes, la méthode de « Bottom-up motion segmentation » et celle du « top-down multiple object tracking ».

La méthode de « Bottom-up motion segmentation » sert à regrouper les trajectoires ponctuelles en se basant sur les signaux de mouvement.
La méthode de « top-down multiple object tracking » quant à elle regroupe les bounding box.

La détection représente ici l'ensemble de points qui appartiennent à des instances d'objets à un moment donné et le suivi est assuré par l'association des données détectés à chaque instant

Gestion des occlusions

Les trajectoires ponctuelles peuvent aider à regrouper les détections de boîtes englobantes du même objet dans des occlusions partielles.

Performances

	MOTA	MOTP	IDS	MT	ML	Hz
MOT16	47.1		370	20.4	46.9	
MOT17	51.2		1851	20.7	37.4	

Multi-Object Tracking with Quadruplet Convolutional Neural Networks [4]

Méthode de tracking basée sur la comparaison de 4 images (Généralisation de réseaux siamois) et d'une perte quadruplet.

Création des 4 réseaux de neurones couplés permettant de mettre un score de similarité entre les 4 images basé sur l'apparence et l'adjacence temporelle.

L'association entre les images et ensuite faite grâce à une propagation de label minimax dans un graphe utilisant le score ainsi calculé.

Gestion des occlusions

L'algorithme compare différentes images, ainsi, il n'est pas contraint par apprendre les caractéristiques propres de l'objet que l'on doit détecter (les réseaux CNN extractant ce paramètre)

Performances

	MOTA	MOTP	IDS	MT	ML	Hz
MOT15	33.8	73.4	70.	12.9	36.9	3.7
MOT16	44.1	76.4	745	14.6	44.9	1.8

Comparaison des méthodes

Méthode	Compréhension	Open Source
DCCRF	+	Pseudo-code disponible
Pseudo code et méthodes expliquées de façon claire dans l'article. Compréhension rapide et simple de l'article. Implémentation envisageable en dépit du fait qu'il ne soit pas open Source.		
MOTDT	++	Oui
Article plutôt clair dans l'ensemble. Utilise néanmoins un filtre de Kalman pour réaliser l'étape de prédiction de objets dans la frame t+. Implémentation envisageable.		
CcC	-	Non
Compréhension globale de l'article et des idées qui en découlent. Mais difficultés à comprendre le détail de l'implémentation de la méthode.		
Quad CNN	--	Non
Difficultés à comprendre le fonctionnement de la méthode.		

Choix de la méthode à utiliser

La solution **MOTDT** est jugée la plus pertinente dans notre situation :

1. Compréhensible
2. Disponible en Open Source
3. Efficace (Fait partie des meilleurs scores disponible sur le benchmark MOT)

En effet, au regard du temps restant sur le projet et dans l'optique d'apporter une solution fonctionnelle au terme du projet, il est intéressant de s'orienter donc vers cette solution.

5. Notre approche (MODT modifié)

5.1. Structure générale

MODT étend le suivi par détection traditionnel en collectant les candidats à partir des sorties de détection et de suivi (tracklet). L'infrastructure du modèle comprend quatre tâches séquentielles, à savoir la classification, la ré-identification, la sélection des candidats et l'association de données.

5.2. Classification

La partie "classification" se base sur une architecture R-FCN pour une classification efficace. Le réseau composé de deux parties, « encodeur » et « décodeur » permet d'avoir à portée de main des

informations difficilement accessible par un classifieur classique. L'étape de classification permet donc de s'assurer que pour chaque ROI (region of interest) l'objet détecté en amont est bel et bien l'objet qu'on a voulu détecter.

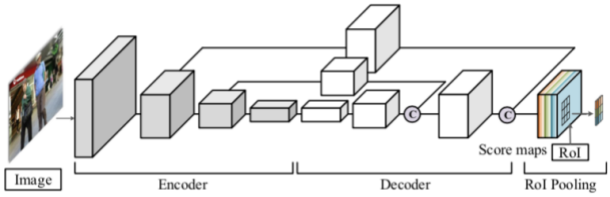


Figure – Architecture du R-FCN

Cette partie n'est pas nécessaire dans notre structure pour deux raisons. La première est que nous travaillons sur des colis (cartons), l'algorithme de détection en amont du réseau rencontrera donc moins de difficultés à détecter des colis. Après plusieurs essaies, Il n'y avait pas assez de changement sur les performances de l'algorithme que ça soit avec ou sans la partie classifications sur les données de colis.
La deuxième raison concerne le temps. En effet c'est intéressant de pouvoir fournir un algorithme optimal en temps.

5.3. Sélection

L'entrée du réseau étant des détections, la sélection est la partie qui sort les probables candidats à suivre et qui sont par la suite associés grâce à un algorithme d'association.
La sélection des probables candidats à suivre se fait à partir des objets suivis de la première frame à la frame t-1(Tracklets), des objets détectés à la frame t ainsi qu'aux positions prédites par le filtre de Kalman des objets dans la frame t. Le filtre de Kalman se base sur les positions des objets à la frame t-1 pour prédire les positions à la frame t.
À partir d'une frame, nous estimons le nouvel emplacement de chaque objets suivis à l'aide du filtre de Kalman. Ces prédictions sont adoptées pour gérer les échecs de détection causés par la variation des propriétés visuelles des objets et l'occlusion dans les scènes encombrées. Mais ils ne conviennent pas au suivi à long terme. La précision du filtre de Kalman pourrait diminuer sur une longue période.

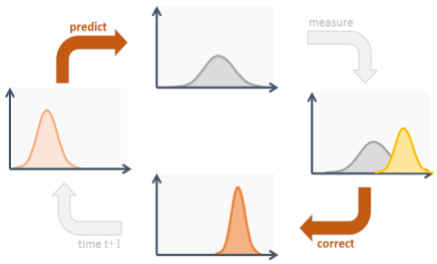


Figure - Cycle d'un filtre Kalman

5.4. Association

L'association des pistes de différents candidats se fait hiérarchiquement et à l'aide de différentes fonctionnalités.
Nous appliquons tout d'abord l'association de données sur les candidats à partir de la détection, en utilisant des représentations d'apparence avec un seuil pour la distance maximale. Ensuite, nous associons les candidats restants à des pistes non associées, basées sur l'IoU (intersection maximum sur union), entre candidats et pistes, avec un seuil. Nous ne mettons à jour que les représentations d'apparence des "tracklets" que lorsqu'ils sont associés à une détection. La mise à jour est effectuée en sauvegardant les fonctions ReID de la détection associée. Enfin, les nouvelles pistes sont initialisées en fonction des résultats de détection restants. Avec l'association de données hiérarchique, il suffit d'extraire les caractéristiques ReID des candidats de la détection une fois par image

5.5. Extraction de features grâce au réseau ReID

L'association des détections faites entre les différentes frames est un élément clé de notre algorithme intervenant dans l'association.
Nous adopterons pour notre algorithme la méthode utilisée par *L. Chen, H. Ai, Z. Zhuang, C. Shang* dans le MOTDT [2].
Cette méthode se base principalement sur la création d'une fonction de similarité entre les différents candidats. Étant donné que l'apprentissage de l'apparence des objets par Deep Learning est la méthode la plus efficace connue à ce jour, nous choisissons donc d'utiliser un réseau de neurones de type GoogLeNet. Ce réseau servira à extraire les vecteurs de caractéristiques des images RVB et permettra donc ensuite de déterminer la similarité entre objets en utilisant la distance entre les caractéristiques obtenues.
Nous utilisons l'architecture réseau proposée dans [5] et l'entraînons sur un dataset de 80000 images. Le réseau H_{Reid} se compose des couches convolutives du GoogLeNet [6] suivie de K branches parallèles formées de couches entièrement connectées (Fully Connected Layers). Vous pouvez vous référer à l'article de Liming Zhao, Xi Li, Jingdong Wang, et Yueting Zhuang [5] pour plus de détails sur cette architecture.

5.5.1 Triplet Loss

L'entraînement du réseau ReID se fait grâce à la méthode du Triplet Loss.

- On fournit en entrée trois images : Deux images du même colis avec des angles de vues différents et une troisième image d'un colis complétement différent $< I_i, I_j, I_k >$
- On calcule ensuite les features des trois images en entrée par passage dans le réseau ReID. Notons $< f_i, f_j, f_k >$ les features map associées aux trois images en entrée, on a $f_i = H_{Reid}(I_i)$
- Ensuite, on calcule les distances euclidiennes entre les différentes features map pour chacun de ses triplets :

$$d_{i,j} = \sum_k |f_{i,k} - f_{j,k}|$$

Avec $f_{i,k}$ et $f_{i,k}$ respectivement les $k^{ème}$ lignes des vecteurs $f_i \; et \; f_j$

- Enfin, on définit une fonction de loss que nous allons minimiser pour entraîner notre réseau :

$$l_{triplet} = \frac{1}{N} \sum_{<I_i, I_j, I_k>} max(d_{i,j} - d_{i,k} + m, 0)$$

Avec N nombre de triplet en entrée et $m > 0$ une marge prédéfinie.
Le maximum ainsi que la marge permettent d'ignorer les triplets tels que $d_{i,j} - d_{i,k} > m$ pour permettre à l'algorithme de mieux discriminer les images "difficiles" (c.à.d. avec des features proches)

Figure - Exemple de triplet d'entraînement (La figure de droite et différente des deux autres)

5.5.2 Base de données utilisées

Pour entraîner le réseau, j'ai utilisé des images de synthèses créées grâce au logiciel Blender.
Les données fournies sont organisées sous forme de scénarios (un scénario par dossier). Chaque dossier est constitué des frames de la vidéo sous format .jpg ainsi que d'un fichier texte décrivant la réalité terrain.

J'ai donc dans un premier temps extrait la totalité des bounding boxes pour créer un dataset d'image de colis. Étant donné que chaque colis apparaît plusieurs fois dans la vidéo (de son entrée à sa sortie du tapis roulant), j'ai toutes les données nécessaires pour l'entraînement.

Finalement, j'ai choisi d'entraîner le modèle sur 424 000 triplets en faisant des batchs de 1 200 triplets.

Vous trouverez un exemple de données sur lesquelles j'ai travaillé dans le dossier dataset sur le git du projet

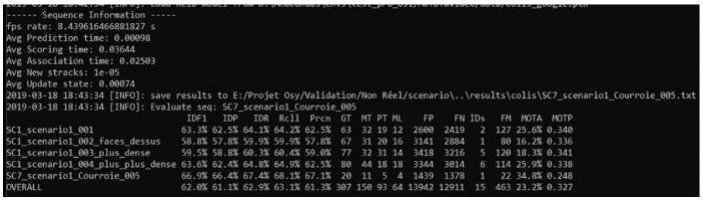
6. Résulats

6.1. Résultats obtenus sur des images de synthèse

Les résultats ont été obtenu sur des données générées grâce au logiciel Blender.

Une légère amélioration s'est fait ressentir en augmentant les données d'entraînement. En effet j'avais à ma disposition une base de données de 50000 images générées par Blender, j'ai entraîné dans un premier temps le modèle sur cette base de données.
N'étant pas satisfait des résultats de cette première simulation, j'ai décidé d'augmenter la base de données en appliquant des rotations de -90° et 90° aux images déjà disponibles ce qui a conduit à une base de données de 423760 images qui a servi à l'entraînement donc à l'amélioration du modèle.

Figure 5 - Exemple de sortie de synthèse réalisée grâce à l'algorithme



```
------ Sequence Information -----
fps rate: 8.439616466881827 s
Avg Prediction time: 0.00098
Avg Scoring time: 0.03644
Avg Association time: 0.02503
Avg New stracks: 1e-05
Avg Update state: 0.00074
2019-03-18 18:43:34 [INFO]: save results to E:/Projet Osy/Validation/Non Réel/scenario\..\results\colis\SC7_scenario1_Courroie_005.txt
2019-03-18 18:43:34 [INFO]: Evaluate seq: SC7_scenario1_Courroie_005
                                  IDF1   IDP   IDR  Rcll  Prcn  GT  MT PT ML    FP    FN IDs   FM  MOTA  MOTP
SC1_scenario1_001                63.3% 62.5% 64.1% 64.2% 62.5%  63  32 19 12  2600  2419   2  127 25.6% 0.340
SC1_scenario1_002_faces_dessus   58.8% 57.8% 59.9% 59.9% 57.8%  67  31 20 16  3141  2884   1   80 16.2% 0.336
SC1_scenario1_003_plus_dense     59.5% 58.8% 60.3% 60.4% 59.0%  77  32 31 14  3418  3216   5  120 18.3% 0.341
SC1_scenario1_004_plus_plus_dense 63.6% 62.4% 64.8% 64.9% 62.5%  80  44 18 18  3344  3014   6  114 25.9% 0.338
SC7_scenario1_Courroie_005       66.9% 66.4% 67.4% 68.1% 67.1%  20  11  5  4  1439  1378   1   22 34.8% 0.248
OVERALL                          62.0% 61.1% 62.9% 63.1% 61.3% 307 150 93 64 13942 12911  15  463 23.2% 0.327
```

Figure 6 - Résultats du tracking sur images de synthèses

6.2. Résultats obtenus sur des images réelles

Le test sur images réelles a été réalisé pour vérifier que notre algorithme est bien fonctionnel avec un vrai flux vidéo.

Les performances sont dans l'ensemble correct. Néanmoins, on remarque des problèmes à droite (en fin de parcours du convoyeur) ainsi que, dans de quelques situations, la création de tracklets à des endroits où il n'y a pas de colis (FP), en particulier en présence d'êtres humains.

Figure 7 - Exemple de sortie image réelle grâce à l'algorithme de tracking et l'algorithme de détections de l'entreprise avec laquelle j'ai travaillé

Conclusion

Le projet de suivi de colis est un projet très complexe, il fait intervenir des notions complexes de visions par ordinateur et de deep Learning.

Je me suis inspirés des articles scientifiques présentant des méthodes de détection et de suivi d'objets, menés une étude approfondie afin de présenter un tableau récapitulatif de potentielles solutions et d'en sélectionner une.
Le code source de la méthode choisie a été analysé afin de cerner l'architecture proposée par la solution pour l'adapter à notre problème, le ré entrainer sur les données de colis et enfin le tester sur plusieurs scenarii de données de colis à ma disposition
Les performances sont acceptables dans l'ensemble. Cependant plusieurs pistes d'améliorations se font ressentir à savoir :

- Meilleure compréhension de la génération des boundings box
- Révision de l'algorithme de matching
- Quantification de l'impact et de l'utilité du ReID
- Meilleur entrainement du réseau :
- Entrainement sur images réelles
- Plus d'epochs sur l'entrainement

References

[1] Deep Continuous Conditional Random Fields with Asymmetric Inter-object Constraints for Online Multi-object Tracking de Hui Zhou, Wanli Ouyang, Jian Cheng, Xiaogang Wang et Hongsheng Li

[2] Real-Time Multiple People Tracking With Deeply Learned Candidate Selection And Person Re-Identification de Long Chen, Haizhou Ai, Zijie Zhuang et Chong Shang

[3] Motion Segmentation & Multiple Object Tracking by Correlation Co-Clustering de Margret Keuper, Siyu Tang, Bjoern Andres, Thomas Brox et Bernt Schiele

[4] Multi-Object Tracking with Quadruplet Convolutional Neural Networks de Jeany Son, Mooyeol Baek, Minsu Cho et Bohyung Han

[5] Deeply-learned part-aligned representations for person re identification de Liming Zhao, Xi Li, Jingdong Wang, et Yueting Zhuang

[6] Going deeper with convolutions de Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Van- houcke, et Andrew Rabinovich

Lien vers données et Programmes

Je mettrais à disposition à travers un lien les scripts développés ainsi que les codes utilisés et un exemple de données réelles et non-réelles (L'ensemble de données d'entrainement coûte beaucoup trop en mémoire)