Détection d'objets en milieu naturel : application à l'arboriculture

Alexandre Dore[1] Michel Devy [1] Ariane Herbulot[12]
[1] CNRS, LAAS, 7 avenue du colonel Roche, F-31400, Toulouse, France
[2] Univ de Toulouse, UPS, LAAS, F-31400, Toulouse, France

Email : {adore, herbulot, devy}@laas.fr

Résumé

Cet article présente une approche de détection de fruits depuis des images acquises par des caméras dans un verger. Le but est d'estimer le nombre de fruits produits par un arbre, ici des pommes de différentes variétés. Nous adaptons une méthode classique, basée sur une classification appliquée sur une fenêtre de résolution variable, déplacée dans toute l'image : ce classifieur doit au préalable être entraîné sur une base d'apprentissage de grande dimension, construite à partir d'images annotées. Cet apprentissage est requis pour les différentes variétés. Pour limiter le temps lié à la construction de ces bases d'apprentissage, nous proposons d'exploiter toujours la même base d'images acquises sur des pommes qu'il est possible de segmenter de manière automatique, typiquement des pommes rouges qui se détachent facilement du feuillage. Nous décrivons les différents classifieurs de type CNN testés pour cette application, exploités en mode prédiction-vérification. Nous comparons cette approche avec une méthode classique de la littérature.

Mots Clef

Détection d'objets, Classification, couleur, CNN, Faster R-CNN, apprentissage.

Abstract

In this article, it is presented an approach for the detection of fruits from images acquired by cameras in an orchard. It is requested to estimae the number of fruits given by a tree, here apples from different species. We adapt the mainstream method, based on a classifier applied on a multi-scale window shifted on all the image : beforehand this classifier must be trained on a large learning database extracted from annotated images. Such a training is required for every apple variety. In order to save time when building these learning database, we exploit always the same data set, acquired on apples that can be automatically segmented, typically red apples that are very salient in the foliage. Different CNN classifiers have been evaluated for this application, executed in a prediction-verification mode. This approach is finally compared with a more classical one.

Keywords

Object detection, Classification, color, CNN, Faster R-CNN, machine learning.

1 Introduction

1.1 Contexte : arboriculture

Le projet PRESTIGE pour lequel est effectué ce travail, a pour but de développer des outils d'assistance aux arboriculteurs en leur donnant un moyen d'estimer le rendement d'un verger, en extrapolant à partir de l'estimation obtenue sur quelques arbres sélectionnés dans ce verger. Cette estimation doit pouvoir se faire à plusieurs étapes de croissance, mais en particulier avant la récolte, cela pour pouvoir planifier les ressources nécessaires (personnel, palox, etc.). Nous devons proposer une approche qui puisse s'adapter à plusieurs fruits (pomme, prune, etc.), mais aussi, pour chacun, à plusieurs variétés. Ici nous nous centrons sur la pomme, dont plus de trente variétés sont produites en France (Golden, Reinette, Gala, Fuji, Granny Smith, etc.).

Même si les arbres sont palissés, même si on acquiert des images des deux côtés de l'arbre, la Vision, quelque soit la bande spectrale, ne donne des informations que sur les objets visibles, donc non occultés par le feuillage ; par ailleurs, les fruits sont souvent groupés en amas (pommes issues d'un m ?me corymbe), qu'il est difficile de segmenter du fait des occultations. Pour détecter les fruits qui sont dans l'arbre ou dans les amas, nous exploiterons un capteur capable de percevoir en volume : un Radar, rigidement couplé au capteur visuel. Actuellement, ce radar est monté sur une platine site-azimut afin de balayer la scène [Henry et al., 2015]. A terme, nous exploiterons donc les données visuelles et les données volumiques.

Dans cet article nous traitons uniquement de la Vision. L'algorithme présenté a deux principaux buts : (1) il doit donner une estimation fiable du nombre de pommes visibles au premier plan et pour chacune, il doit déterminer la probabilité que ce soit une pomme isolée ou un amas de pommes ; puis, (2) il doit donner la position précise des fruits pour les détections les plus sûres, donc en minimisant les fausses détections.

Par la suite, ces positions seront exploitées pour pointer le radar sur les zones où sont les pommes, cela afin de caractériser le signal réfléchi selon que cela soit une pomme isolée ou un amas de pommes. Les résultats obtenus durant cette phase, seront ensuite exploités pour permettre un comptage par le biais du radar, comptage complémentaire de celui obtenu par Vision puisqu'il se fait dans le volume de l'arbre, sans tenir compte du feuillage.

La détection d'objets genériques dans des images, est une thématique ancienne de la Vision par Ordinateur, qui a connu des progrès impressionnants depuis les années 2000, grâce à des approches fondées sur l'apprentissage de modèles caractéristiques de classes d'objets ; le challenge PASCAL VOC [Everingham et al.,] a permis de mesurer chaque année cette évolution de 2005 à 2012. Sont apparus d'abord des descripteurs (ondelettes de Haar, ensemble de points d'intérêt, HOG, BoW, etc.) permettant de caractériser des régions d'intérêt, puis des méthodes de classification (AdaBoost, Random Forest, SVM, etc.) entraînées pour détecter la présence d'objets dans des régions déplacées en toutes positions de l'image requête.

Depuis 2010 environ [Krizhevsky et al., 2012], ces approches sont concurrencées par les méthodes fondées sur les réseaux de neurones de convolution (CNN), dont les capacités d'apprentissage sont bien supérieures, à condition de disposer de bases d'apprentissage de très grandes dimensions, et de moyens de calcul très puissants : ici aussi il a été proposé d'exploiter des approches à plusieurs étapes, en faisant varier la complexité des architectures des CNN.

Cet article traite de la détection d'objets naturels -fruits avant récolte- dans des images acquises en extérieur dans des vergers. Les fruits doivent être détectés avec des conditions d'illumination très variables et avec des occultations dues au feuillage. Avant d'introduire l'approche proposée, nous donnons quelques éléments de contexte.

1.2 Approche visuelle pour détecter des pommes

Nous avons d'abord testé de nombreuses approches, en tirant parti des méthodes existantes, en particulier dans OpenCV : segmentation par la couleur, détection par les gradients ou par les contours, combinaison de classifieurs (AdaBoost, SVM, ANN) appliqués sur des régions d'intérêt ou sur des pixels, etc. Ces diverses méthodes ne permettaient pas d'atteindre des taux de classification visés dans le projet, supérieurs à 80%, en particulier pour le comptage des pommes vertes, puisqu'il est plus difficile de les segmenter du fait du feuillage.

Nous avons ensuite évalué les approches dite de Deep Learning, avec des classifieurs CNN. Les difficultés majeures concernent (1) l'apprentissage, sachant qu'il n'existait pas de bases disponibles avec des images de fruits en verger, et (2) le choix de l'architecture du réseau, sachant que nous ne disposons pas dans notre équipe, de fermes de calcul, permettant de multiplier les entraînements en faisant varier les paramètres du réseau.

Nous décrivons ci-après comment nous avons surmonté ces difficultés. La section 2 présente quelques travaux dédiés à la segmentation des fruits dans des arbres. En section 3 nous présentons et comparons les deux approches CNN que nous avons évaluées : Faster R-CNN et YOLO. La section 4 décrit notre approche, en particulier, comment nous avons construit une base d'apprentissage avec un temps limité pour annoter des images. Nous présentons une comparaison entre deux approches de détection : celle fondée sur Faster R-CNN, et une méthode plus classique, qui exploite AdaBoost pour la prédiction, puis ANN pour la vérification. La section 5 décrit des traitements complémentaires, requis pour que les détections produites par ces méthodes soient exploitées par le Radar. Enfin la section 6 résume nos contributions et évoque nos futurs travaux.

2 Etat de l'art

L'une des difficultés de la comparaison des différentes méthodes de détection de pommes, est qu'il n'existe pas de base de données d'images acquises sur des pommes dans un verger. En général les images utilisées pour la détection de fruits ne les représentent pas dans des arbres, mais de façon isolée, ce qui change la nature du problème.

Plusieurs algorithmes de détecteurs d'objets ont été testés dans le cadre de la détection de pommes dans des vergers avec des résultats plus ou moins satisfaisants.

La méthode proposée par [Wachs et al., 2010] repose sur une détection d'objets utilisant l'algorithme de Viola et Jones [Viola and Jones, 2004] appliqué sur des images multi-spectrales (visible achrome et thermique) acquises séparément sur un arbre. Les régions classées positives par AdaBoost, sont ensuite analysées par plusieurs réseaux de neurones votant pour confirmer ou infirmer ces premières détections.

D'autres méthodes s'appuyant sur une segmentation par la couleur des pixels donnent de bons résultats si les images sont de très bonne qualité [Vaysse et al., 2012] [Linker et al., 2012], ce qui nécessite en général des techniques de photographie inutilisable de manière automatique. La méthode MECAVISION [Vaysse et al., 2012] a plus particulièrement été testée, celle ci reposant sur une détection des pommes visibles grâce au gradient et un apprentissage de la couleur des objets détectés pour classer le reste de l'image. Le principal problème de cette méthode est le fait que la première détection suppose que une ou plusieurs pommes soient assez visibles, mais aussi que toutes les pommes soient éclairées de manière uniforme ce qui est impossible sans utiliser un éclairage puissant et bien positionné. Cette condition est rédhibitoire dans notre contexte applicatif.

Certaines méthodes utilisent aussi une classification en utilisant la texture [Zhao et al., 2005]. Mais les résultats que nous obtenons sur nos données ne sont pas satisfaisants. La texture n'est pour nos images pas un critère suffisant pour discriminer une feuille d'une pomme verte. Certaines pommes ayant une texture plus forte que certaines feuilles

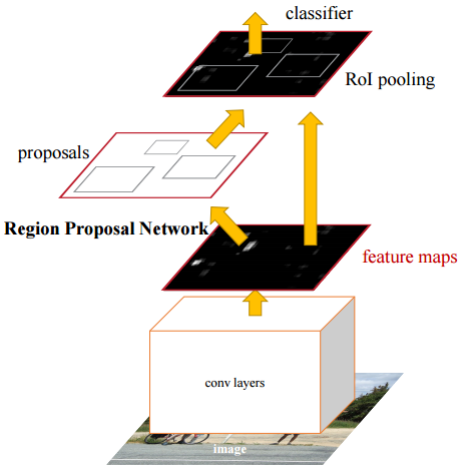


FIGURE 1 – Réseau de neurones de convolution (pris de [Ren et al., 2015])

et inversement.
S.Bargoti et al. [Bargoti and Underwood, 2016] utilisent une méthode de détection utilisant des réseaux de neurones de convolution. Elle se base sur l'algorithme Faster R-CNN résumé ci-dessous [Ren et al., 2015]. Cette méthode permet de différencier différents fruits (pommes, mangues et amandes), pommes qui dans ce contexte sont des pommes de variété virant au rouge (Pink lady et Nicoter).

3 Réseaux de neurones de convolution pour la détection d'objets

Dans cette section nous détaillons deux outils disponibles proposés récemment exploitant les réseaux de neurones de convolution pour la détection d'objets : Faster R-CNN et YOLO. Nous avons choisi ces outils, car YOLO permet d'obtenir les meilleurs résultats sur les données VOC2007 et VOC2012 et Faster R-CNN est une des méthodes CNN les plus utilisées à ce jour.

3.1 Faster R-CNN

Le premier détecteur testé pour la détection de pommes est l'algorithme créé par S.Ren et al [Ren et al., 2015] qui repose sur une détection entièrement faite avec des réseaux de neurones de convolution (figure 1) :

- un premier réseau de neurones de convolution prend en entrée une image de taille quelconque et donne en sortie des régions dans lesquelles pourraient se trouver les objets à détecter.
- le second réseau prend en entrée les régions proposées par le premier réseau et recherche si elles contiennent l'objet à détecter.

Pour faire leur détection, S.Bargoti et al. utilisent un modèle développé par K. Simonyan et al [Simonyan and Zisserman, 2014].
Dans notre cas les limitations matérielles nous forcent à utiliser des architectures de réseau de taille limitée.

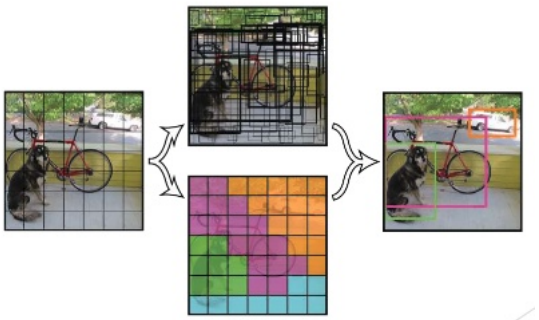


FIGURE 2 – Algorithme YOLO (pris de [Redmon et al., 2015])

Dans la suite on choisit de prendre l'architecture décrite en [Chatfield et al., 2014], modèle dérivé de l'architecture de K.Simonyan et al.

3.2 YOLO

Le second détecteur testé est l'algorithme développé par J.Redmon et al. [Redmon et al., 2015]. Cet algorithme repose sur deux étapes qui sont appliquées sur des images de taille prédéfinie lors de l'apprentissage (figure 2) :

- une détection d'objets opérée par des réseaux de neurones de convolution ;
- un quadrillage de l'image où l'on prédit la classe de l'objet s'il existe (dans notre cas soit une pomme soit rien).

L'avantage de cette prédiction est qu'elle peut se faire indépendamment de la première détection mais qu'elle ne cible pas les objets en particulier ; il est donc plus difficile de détecter les plus petits objets ainsi que les objets se chevauchant.
Nous avons testé aussi la dernière version YOLO9000 proposée en [Redmon and Farhadi, 2016]. Deux architectures de réseau CNN sont décrites dans ce papier, l'une permettant une classification plus rapide mais avec une perte sur l'erreur de détection. Les contraintes matérielles nous ont imposés pour le moment à n'utiliser que la version la moins performante.

3.3 Comparaison des méthodes de détection

L'une des principales différences entre YOLO et Faster R-CNN est le temps de calcul, YOLO permet d'avoir une détection de 37 images par seconde pour une image de 445x445x3 alors que Faster R-CNN permet d'avoir seulement 5 images par seconde (sur les images prises dans des vergers). De plus sur les data sets VOC2012 et VOC2007, YOLO semble donner de meilleurs résultats.

4 Détection des pommes et comptage

Notre approche consiste en plusieurs étapes ; la première est de détecter les pommes sur des images acquises dans les vergers, par une des méthodes de détection d'objets vues précédemment ; ensuite nous améliorons ces résultats en

caractérisant les pixels acquis sur les pommes, puis en estimant une position dans l'image plus précise pour chaque pomme détectée, ainsi que leur taille moyenne.
Cette section décrit la premi ?re étape : nous montrerons l'efficacité des méthodes décrites plus haut pour détecter les régions de l'image dans lesquelles sont les pommes ; rappelons que dans notre contexte, les vergers produisent plusieurs variétés de pommes ; dans les cas les plus difficiles, les couleurs des pommes peuvent se confondre avec le décor. Nous montrerons dans la section suivante, qu'il est possible d'améliorer la qualité de la détection en estimant plus précisément la taille et la position des pommes dans les régions retenues dans la premi ?re étape.

Création de données d'entraînement.
L'un des principaux problèmes lors de l'entraînement de ces détecteurs (Faster R-CNN, YOLO et le détecteur inspiré par [Wachs et al., 2010]) est le nombre d'imagettes de pomme nécessaire pour l'apprentissage des classifieurs, ainsi que la qualité de des imagettes : généralement les pommes dans ces imagettes, doivent être centrées et de la bonne taille.
Dans notre cas, les données utilisées pour l'entraînement sont composées d'une cinquantaine d'images annotées contenant en tout près de 800 pommes (80% de pommes vertes et 20% de pommes rouges). Ces 800 imagettes contenant des pommes, sont séparées en deux groupes, 70% pour l'entraînement et 30% pour la validation. Pour augmenter artificiellement ce nombre, plusieurs méthodes existent déjà , la plupart consistant à faire des transformations d'images en changeant l'orientation ou en appliquant un changement d'échelle.
Dans notre cas, il est possible d'augmenter artificiellement le nombre de pommes vertes annotées pour l'apprentissage ; ces pommes sont les moins facilement différenciables. En effet la principale difficulté de la détection vient surtout du fait que la couleur des pommes vertes ne diffère pas réellement de la couleur des feuilles aux alentours.
Pour augmenter le nombre de données on peut donc transformer les pommes rouges en pommes vertes. Pour cela il suffit pour chaque pomme rouge des données d'entraînement, d'extraire les pixels de pommes des données en entraînement (une segmentation en chrominance rouge suffit) et de remplacer les chrominances rouge et bleu par celles d'une pomme verte prise au hasard dans l'ensemble des pommes vertes des données d'entraînement (voir figure 3). Avec ces différentes opérations le nombre d'imagettes de pommes disponibles à la fois pour l'apprentissage et pour l'évaluation, est passé de 800 à environ 4000.

Comparaison des principales méthodes de détections.
Il a été choisi de comparer les résultats obtenus par les outils Faster R-CNN ou YOLO avec une méthode inspirée de [Wachs et al., 2010], celle-ci donnant les meilleurs résultats avant l'exploitation des réseaux de neurones convolutifs. Les résultats de cette comparaison sont présentés en table 1 et figure 4.

(a) Pommes rouges (b) Pommes vertes

FIGURE 3 – Transformation de pommes rouges en pommes vertes pour la base d'apprentissage

La méthode de référence n'est pas la même que celle décrite dans [Wachs et al., 2010], car il était impossible pour nous de disposer d'une image thermique de la scène, comme le font J.P.Wachs et al. La première détection se fait donc seulement sur une image en niveaux de gris et non pas par une fusion multi-spectrale visible-infrarouge.
De plus la confirmation ne se fait plus par vote de chaque réseau de neurones mais par une moyenne de leur score de détection. Et ces réseaux prennent en compte le gradient (angle et magnitude) des pixels de l'image pour la classification en plus des données couleurs des pixels, ce qui à l'entraînement permet d'améliorer les performances, en diminuant le taux de fausse détection d'environ 5% et le taux de non détection d'environ 10%.
Il est aussi possible d'améliorer le nombre de pommes détectées en abaissant le seuil de détection de la classification en cascade, mais cela implique une multitude de détection pour la plupart des pommes, et une augmentation importante du taux de fausse détection.
On remarque que la méthode utilisant l'algorithme de détection Faster R-CNN donne globalement de meilleurs résultats, mais qu'il ne faut tout de même pas oublier le fait que la détection par classification en cascade n'utilise pas d'image infrarouge comme préconisé par J.P.Wachs et al.
Lors de la détection utilisant l'algorithme Faster R-CNN le réseau permettant de faire la première détection ne parvient pas à détecter 7% des pommes (faux négatifs), et sur chaque image retourne la position des 300 fenêtres contenant avec le plus de certitude un objet. Le deuxième réseau va invalider à tord certaines de ces détections : 13% des pommes ne seront pas détectées. Il est possible d'améliorer les résultats en ayant une estimation du rayon des pommes, car cela permet de séparer les groupes de pommes.
On peut remarquer que Faster R-CNN ne parvient pas à discerner toutes les pommes de l'amas présent dans la figure 4.a, et discerne moins bien les pommes sombres que la méthode de référence. La majorité des fausses détections vient du fait qu'il n'arrive pas à faire la différence entre une pomme et certaines feuilles trop grosses.
Les taux de fausse détection pour les pommes rouges, qui sont plus faciles à détecter, tombent à 2% pour Faster R-

TABLE 1 – Comparaison entre méthodes de détection des pommes.

méthode de détection	taux de fausse détection	taux de non détection
Faster R-CNN	0.120	0.203
Ondelettes de Haar et réseaux de neurones (J.P.Wachs et al.)	0.172	0.488
YOLO	$\simeq 0.3$	$\simeq 0.6$

CNN, et à 7.5% pour l'algorithme inspiré de J.P.Wachs et al., alors que les taux de non détection restent les mêmes.

Remarquons que pour l'algorithme YOLO, la détection est de mauvaise qualité : plus de 60% de pommes ne sont pas détectées sur des images prises à 1 mètre et de plus, nous trouvons 30% de fausses détections.

L'un des problèmes de cette méthode de détection est qu'en général pour une pomme il y aura une multitude de détections qui ne peuvent pas être associées, celles-ci ne se recoupant pas toujours. De plus les détections sont parfois décentrées par rapport aux pommes présentes sur l'image, ce qui peut être dû à un trop petit nombre de données d'entraînement.

L'une des explications possibles de cette différence entre YOLO et Faster R-CNN est le fait que Faster R-CNN lors de la classification, s'adapte à la détection faite : elle est donc spécifique à l'objet détecté, contrairement à YOLO qui classe une zone entière, et donc va avoir tendance à classer comme pomme un objet détecté dans le voisinage de celle-ci.

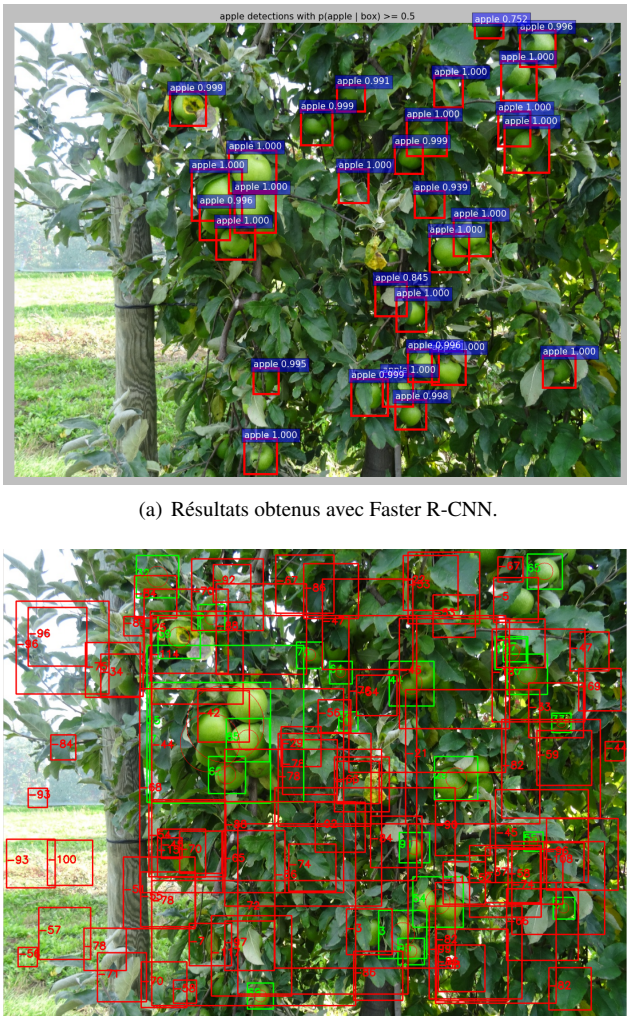
5 Amélioration des détections

A ce stade Faster R-CNN donne des fenêtres dans lesquelles une ou plusieurs pommes ont été détectés. Pour la suite des travaux (utilisation d'un radar pour affiner les détections ou estimer la croissance des fruits), il sera utile de pouvoir déterminer la position et la taille des pommes dans ces fenêtres le plus précisément possible, ainsi que d'avoir des informations sur leur entourage.

5.1 Caractérisation du voisinage des détections

Pour affiner le résultat du classifieur, il est utile de pouvoir savoir quels pixels dans les fenêtres issues de la détection, sont des pixels de pommes, cela afin de caractériser le voisinage de celle-ci, mais aussi pour le recentrage. Pour faire cette classification niveau pixel, on effectue un apprentissage de la couleur d'une pomme dans l'image étudiée. Les données d'apprentissage sont les pixels détectés comme étant des pommes lors de la première détection (le seuil étant plus élevé pour éviter de prendre en compte des pixels mal classés par le premier détecteur), le reste des pixels de l'image étant considéré comme des pixels de 'non pomme'.

L'un des problème de cet apprentissage est que l'on ne connaît pas de pixels n'appartenant pas à la classe pomme avec certitude, if faut donc pouvoir laisser lors de la phase d'apprentissage une certaine liberté quant au nombre de pixels mal classés. Une discrimination quadratique semble adaptée pour ce problème (figure 5).



(a) Résultats obtenus avec Faster R-CNN.

(b) Résultats obtenus avec la méthode de référence.

FIGURE 4 – Comparaison entre méthodes de détection.

Le résultat de la segmentation peut être utilisé pour réduire le nombre de faux positifs et ne garder que les pommes visibles. De plus le recentrage des détections permet d'éviter de prendre en compte des pixels n'appartenant pas à la classe "pomme".

5.2 Recentrage des détections sur les pommes

Afin de pointer le radar sur une pomme détectée par vision, il est important d'avoir des détections centrées sur l'objet.

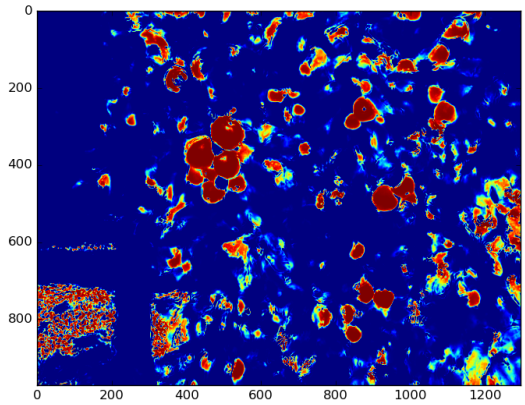


FIGURE 5 – Résultat de la classification en appliquant une discrimination quadratique (les pixels étant en rouge sont considérés comme des pixels de pommes)

Ces fenêtres pourront servir aussi pour améliorer la base exploitée pour l'apprentissage de la couleur des pixels de pomme ; il est donc important quelles soient le plus fiable possible.
Une des façons de corriger ce décentrage, est d'exploiter le fait que les pommes ont une forme circulaire : on peut donc en théorie calculer le centre et le rayon de la pomme pour recadrer l'image. Deux méthodes ont été testées, toutes deux consistent à faire une régression circulaire sur les points de contours. La première tente de minimiser la somme des erreurs quadratiques faites par la régression alors que la seconde minimise le produit de ces erreurs.

Minimisation de la somme des erreurs.
Ici la fonction à minimiser est :
$E[a,b,r] = \sum(\sqrt{(X_i-a)^2+(Y_i-b)^2}-r)^2$
avec (a,b) le centre du cercle à déterminer, r son rayon, et (X_i, Y_i) les coordonnées du i^{me} point de contour.
Plusieurs méthodes de descente de gradient permettent d'obtenir de bons résultats. Mais une méthode plus fiable est présentée en annexe. Cette minimisation donne une bonne approximation du résultat avec un faible temps de calcul.
Mais cette méthode est très sensible aux occultations ; l'estimation se dégrade si plusieurs pommes se touchent ou si elles sont occultées par une feuille ou une branche.

Minimisation du produit des erreurs.
Une solution pour pallier ce problème lié aux occultations est de non pas minimiser la somme des erreurs mais le produit de celles-ci, la fonction à minimiser devient donc :
$E[a,b,r] = \prod(\sqrt{(X_i-a)^2+(Y_i-b)^2}-r+1)^2$
Mais cette fonction peut prendre des valeurs aberrantes, trop grandes pour être prises en compte dans la plupart des langage de programmation. Il est donc préférable d'utiliser :
$E[a,b,r] = \sum log((\sqrt{(X_i-a)^2+(Y_i-b)^2}-r+1)^2)$
Cette méthode permet d'avoir des résultats plus fiables dans le cas où les pommes ont un contour circulaire mais va avoir tendance à donner de mauvais résultats dans le cas contraire.
Pour déterminer le minimum de cette fonction une descente de gradient ne peut être utilisée, car la fonction à minimiser possède en général de nombreux minima locaux.
La solution retenue est d'appliquer un algorithme d'optimisation métaheuristique : la recherche harmonique [Jacquelin, 2009], décrite en annexe. Cette méthode prend plus de temps que la descente de gradient (environ 50 fois plus de temps), mais donne de meilleurs résultats.

Comparaison des deux méthodes de recadrage.
Le principal avantage de la minimisation de la somme des erreurs quadratiques est qu'elle est plus rapide que la seconde minimisation mais elle donne des résultats de moins bonne qualité, en particulier sur des pommes qui sont occultées, comme cela apparaît clairement en figure 7. En effet en minimisant la somme des erreurs la solution donne un cercle qui va minimiser l'écart entre le cercle et tous les points pris en compte : il suffit donc d'avoir des points aberrants pour dégrader le résultat (figure 7, courbe violette).
La figure (figure 6(a)) représente une fenêtre contenant une pomme occultée par le feuillage, malgré tout détectée par la méthode Faster R-CNN ; cette fenêtre est non centrée sur la pomme ; nous montrons aussi la meilleure approximation de la forme de la pomme. Les deux autres figures (figure 6(b)) et (figure 6(c)) représentent deux résultats de recentrage sur le centre du cercle approximant la forme de la pomme ; le recadrage est meilleur ? droite, car le cercle approxime mieux la forme de la pomme.
La méthode à retenir en pratique dépendra du temps de calcul souhaité, car même si la minimisation de la somme des erreurs donne de moins bons résultats, sa rapidité d'estimation en fait une solution intéressante.
Des méthodes plus classiques de contours actifs donnent des résultats souvent médiocres, la texture et donc le gradient pouvant varier au sein d'une même pomme et parfois ténue en particulier dans les zones d'ombre. Par ailleurs, ll'initialisation peut poser problème ; en effet lorsque la pomme est partiellement occultée par une feuille, le contour risque de converger sur la feuille, et non sur la pomme.

6 Conclusions

Nous avons présenté une approche adaptée à la détection des fruits dans des arbres. Nous montrons que le classifieur basé sur Faster R-CNN donne de meilleurs résultats que les méthodes plus classiques, exploitant un classifieur de type Viola-Jones. De plus nous avons mis au point des algorithmes permettant d'améliorer ces détections en réestimant la position des détections et en essayant d'en déduire la position des pixels de pommes.
Nous avons montré sur quelques arbres, qu'il est possible d'avoir une estimation de la production et ce quelque soit

(a) Image avant recadrage, le cercle bleu correspondant à la meilleure estimation

(b) Après recadrage avec somme des erreurs quadratiques

(c) Après recadrage avec produit des erreurs quadratiques

FIGURE 6 – Comparaison des méthodes de recadrage.

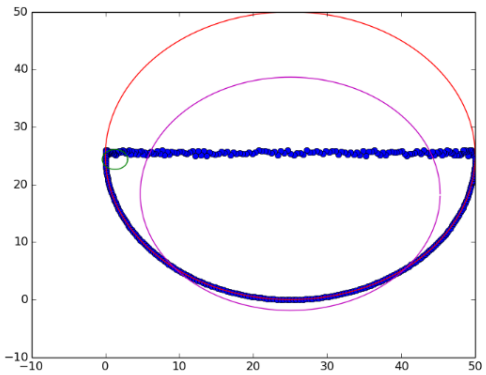


FIGURE 7 – Comparaisons des différentes méthode d'estimation d'un cercle occulté : en bleu les points à interpoler, en violet en utilisant la minimisation de la somme des erreurs, en vert en minimisant le produit des erreurs par descente de gradient et en rouge en minimisant le produit des erreurs avec par métaheuristique

la variété de la pomme, mais il reste ? faire des tests plus complets, en particulier en tirant parti de la fusion Radar-Vision.

En effet, la vision ne perçoit que la surface des objets ; pour estimer le rendement d'un arbre, il convient de combiner la Vision avec un capteur capable de percevoir dans le volume. Nous avons donc une collaboration avec une équipe qui développe un Radar pour des applications agricoles.

Deux stratégies de fusion sont étudiées : d'abord dans une phase d'initialisation, une approche séquentielle "Vision, puis Radar" afin de caractériser le signal réfléchi par une pomme isolée ou par un amas de pommes ; ensuite dans une phase d'exploitation, une fusion "lâche" au niveau géométrique pour valider avec les deux capteurs les détections des objets en surface de l'arbre, sachant que seul le Radar pourra détecter les fruits masqués par le feuillage. Nous avons ici décrit les traitements de Vision nécessaires pour la fusion séquentielle : nos travaux actuels, en collaboration avec les radaristes, concernent la fusion lâche.

Remerciements

Ces travaux ont été réalisés dans le cadre du projet PRESTIGE, financé par la région Occitanie et par des fonds FEDER.

7 ANNEXE

7.1 Minimisation de la somme des erreurs quadratiques

Pour pouvoir déterminer si une partie de contour ressemble à un arc de cercle on peut calculer le centre et le rayon permettant d'avoir un cercle le plus proche des points sélectionnés.

On cherche ici à minimiser l'erreur quadratique. On cherche donc (a,b,r) tel que :$\sum(x_i - a)^2 + (y_i - b)^2$ soit minimal avec x et y les coordonnées des points du contour.
Pour approcher le centre du cercle et son rayon on peut résoudre le système $Ax = B$ avec :

$$x = \begin{pmatrix} a \\ b \\ s \end{pmatrix} \; avec \; s = a^2 + b^2 + r^2$$

$$A = \begin{pmatrix} 2\sum x_i^2 & 2\sum x_i y_i & \sum x_i \\ 2\sum x_i y_i & 2\sum y_i^2 & \sum y_i \\ 2\sum x_i & 2\sum y_i & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} \sum(x_i^3 + x_i y_i^2) \\ \sum(x_i^2 y_i + y_i^3) \\ \sum(x_i^2 + y_i^2) \end{pmatrix}$$

En effet en considérant la fonction d'erreur : $E(a,b,r) = \sum E_i(a,b,r)$ avec $E_i(a,b,r) = (r^2 - (x_i - a)^2 + (y_i - b)^2)^2$

On peut chercher les minimum, donc là où la dérivée s'annule, soit (A,B,R) le point de l'espace des paramètres ou les dérivées partielles s'annulent. On a donc :

$\frac{\delta E}{\delta a} = -2\sum x_i E_i + 2A\sum E_i = 0$
$\frac{\delta E}{\delta b} = -2\sum y_i E_i + 2B\sum E_i = 0$
$\frac{\delta E}{\delta r} = -2R\sum E_i = 0 \; or \; R > 0 \; donc \; \sum E_i = 0$
et en développant on obtient bien le système ci-dessus.

Une solution exacte peut être trouvée en utilisant la méthode décrite en [Jacquelin, 2009].

7.2 Minimisation du produit des erreurs par recherche harmonique

L'algorithme de recherche harmonique est un algorithme d'optimisation permettant de trouver le minimum d'une fonction. Elle fait parti des méthodes Métaheurstiques qui sont des solutions itératives pour résoudre des problèmes d'optimisation.

En général ces méthodes s'initialisent en prenant aléatoirement un certain nombre de points initiaux, qui vont converger vers des minimums locaux. A chaque itération ces points vont prendre en compte les informations des autres points (valeur de la fonction, position, dérivées etc...) pour affiner la recherche du minimum global.

La méthode de recherche d'harmonie se décompose en 5 étapes :

- la première initialise les constantes : critère d'arrêt, nombre de solutions de départ HMS (Harmony Memory Size), taux de sélection $HMCR$, taux d'ajustement par et déplacement δ ;
- la deuxième étape génère aléatoirement des solutions au problème. On calcule pour chaque solution la valeur de la fonction que l'on stocke dans la matrice "Mémoire" HM

$$HM = \begin{bmatrix} x_1^1 & \dots & x_1^1 & f(x^1) \\ \vdots & \ddots & \vdots & \vdots \\ x_1^{hms} & \dots & x_1^{hms} & f(x_{hms}) \end{bmatrix}$$

  et on crée une nouvelle solution x' pour chaque composante. Cette solution dépend du taux de sélection et du taux d'ajustement ; on prend comme valeur :
  - avec une probabilité $HMCR$, on extrait aléatoirement une des composantes des anciennes solution $x'_i \leftarrow x_i^{int(u(0,1)*HMS)+1}$.
  - avec une probabilité $1 - par$ on choisit aléatoirement la composante.
- l'étape trois consiste à ajuster le nouveau candidat en lui soustrayant δ avec une probabilité par
- la quatrième étape consiste à comparer la nouvelle solution à la plus mauvaise de l'étape 2 et elle la remplace si elle donne de meilleurs résultats ;
- la dernière étape regarde le critère d'arrêt. S'il n'est pas satisfait on retourne à l'étape 3.

Le programme utilisé pour les tests est disponible en Python [Geem et al., 2001] et est une implémentation de la méthode décrite en [Fairchild, 2012].

Références

[Bargoti and Underwood, 2016] Bargoti, S. and Underwood, J. (2016). Deep fruit detection in orchards. *CoRR*, abs/1610.03677.

[Chatfield et al., 2014] Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details : Delving deep into convolutional nets. *CoRR*, abs/1405.3531.

[Everingham et al.,] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http ://host.robots.ox.ac.uk/pascal/VOC/.

[Fairchild, 2012] Fairchild, G. (2012). Harmony search. *https ://github.com/gfairchild/pyHarmonySearch*.

[Geem et al., 2001] Geem, Z., Kim, J., and Loganathan, G. (2001). A new heuristic optimization algorithm : Harmony search. *Simulation*, 76(2) :60–68.

[Henry et al., 2015] Henry, D., Pons, P., and Aubert, H. (2015). 3d microwave imaging system for the remote detection and reading of passive sensors. In *European Microwave Week (EuMW), Paris (France)*.

[Jacquelin, 2009] Jacquelin, J. (2009). Regressions coniques, quadriques, circulaire et spherique. *https ://fr.scribd.com/document/14819165*.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Linker et al., 2012] Linker, R., Cohen, O., and Naor, A. (2012). Determination of the number of green apples in rgb images recorded in orchards. *Computers and Electronics in Agriculture*, 81 :45–57.

[Redmon et al., 2015] Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You only look once : Unified, real-time object detection. *CoRR*, abs/1506.02640.

[Redmon and Farhadi, 2016] Redmon, J. and Farhadi, A. (2016). YOLO9000 : better, faster, stronger. *CoRR*, abs/1612.08242.

[Ren et al., 2015] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn : Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

[Vaysse et al., 2012] Vaysse, P., Reynier, P., Mathieu, V., Roche, L., Keresztes, B., Lavialle, O., and Guizard, C. (2012). L'éclaircissage du pommier, un nouveau pilotage nommé mecavision. *Publication CTIFL*, 287.

[Viola and Jones, 2004] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2) :137–154.

[Wachs et al., 2010] Wachs, J. P., Stern, H. I., Burks, T., and Alchanatis, V. (2010). Low and high-level visual feature-based apple detection from multi-modal images. *Precision Agriculture*, 11(6) :717–735.

[Zhao et al., 2005] Zhao, J., Tow, J., and Katupitiya, J. (2005). On-tree fruit recognition using texture properties and color data. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.