Detection et Segmentation d'image à l'aide de la méthode deeplearning

Claire Hamonet, Hugo Joby, Flavien Ronteix–Jacquet

Abstract

L'objectif de ce travail est de comprendre la problématique de la détection et la segmentation des objets dans une image, d'où la problématique,

Comment détecter, reconnaître et segmenter les objets présents dans une image ?

Pour ce faire nous avons réalisé une étude bibliographique et analysé les dernières recherches dans ce domaine porteur qu'est l'intelligence artificielle.

1. Introduction

. Assistants personnels, voitures, recommandations... "l'intelligence artificielle" est aujourd'hui omniprésente. Pourtant, à l'origine, l'un des premiers domaines où l'intelligence a été appliquée fut la vision par ordinateur. En effet, beaucoup de tâches humaines sont facilement automatisable mais la vision quand à elle, requiert un apprentissage (les formes, les couleurs, etc...) ainsi qu'une instinctivité, chose que la machine ne possède pas (dû moins pas encore). Certains traits propre à l'intelligence comme la faculté de connaître, d'apprendre et de comprendre ont été nécessaire au domaine de la vision et plus particulièrement à la reconnaissance et la détection de l'environnement.

. Dans notre étude, nous allons nous restreindre aux technologies d'intelligence artificielle pour la détection et la segmentation des objets dans une image. Le document ci-dessous nous permet de voir clairement les différents types de vision. La plus ancienne méthode est la classification, c'est ce que faisait Y. LeCun avec les premiers réseaux neuronaux. L'objectif est de dire quel objet est le plus présent dans la scène. Il y a ensuite la segmentation sémantique, l'objectif est seulement de découper l'image en "région d'objet" et de donner un label à chaque région, c'est une des solutions que nous étudierons. Une autre solution est la détection, il n'est plus question de se baser sur des pixels

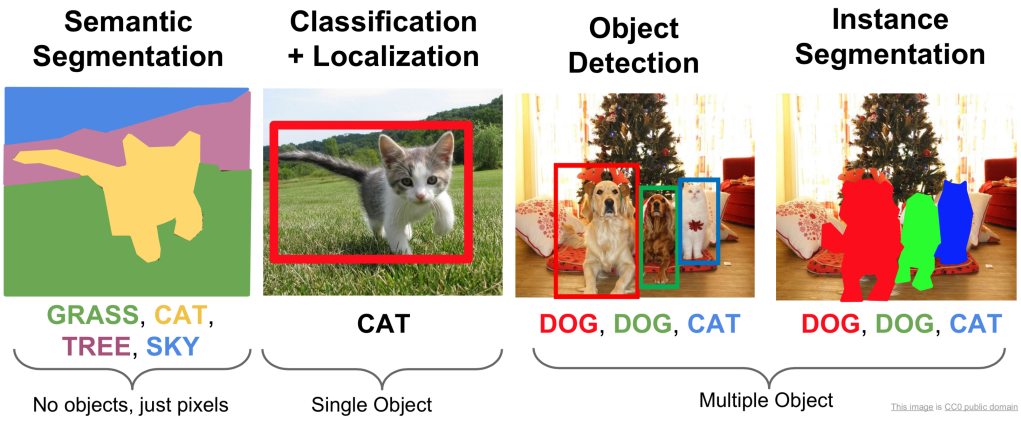


Figure 1: Différent type d'analyse d'image

mais sur des fenêtres ou des rectangles pour chacun des objets reconnus dans l'image. Il existe une dernière solution : la segmentation d'instance. C'est la combinaison des 2 solutions précédentes (non étudiée ici).

. Ainsi, nous allons dans cette étude mener un état de l'art des méthodes de détection et de segmentation. Puis nous analyserons une nouvelle technologie prometteuse en comparant avec ce qui se faisait il y a seulement 3 ans et ainsi montrer l'évolution spectaculaire de la recherche. Enfin nous concluerons sur l'intérêt de ces recherches, les applications, les perspectives d'avenir ainsi que notre opinion sur le sujet.

2. L'état de l'art

2.1. Introduction au deeplearning

. Le deep-learning ou apprentissage profond, est une méthode particulière du machine learning qui a pour but l'apprentissage automatique d'une machine à partir de modèles. Le deep-learning se base sur des couches de neurones (réseau neuronaux) empilés en modèle qui vont se faire une "représentation" du monde à la suite d'un apprentissage.

. Le but de réseau neuronal est d'imiter le fonctionnement du cerveau humain avec des neurones qui interagissent ensemble par des synapses pour pouvoir copier son incroyable capacité d'abstraction mais surtout d'apprentissage avec une amélioration continuelle due à l'expérience acquise.

. Ainsi, l'on peut distinguer 2 types d'apprentissage : supervisé (on donne des exemples déjà classés à la machine et elle doit apprendre de ce données) et non-supervisé (elle doit trouver toute seule des structures dans les données fournies ce qui est beaucoup plus difficile). Dans notre étude, on se basera uniquement sur un apprentissage supervisé. L'apprentissage non-supervisé est l'objectif finale de la recherche en intelligence artificielle et elle n'est à ce jour pas encore implémentée.

2.2. Un dataset en commun, le PASCAL VOC 2012

. Parlons donc de dataset. Pour tester et évaluer un modèle, il est impératif de disposer d'une collection de données. Cette collection sera reprise à chacun des tests afin de pouvoir comparer au mieux les résultats des différentes méthodes appliquées. Or, à la suite de nos recherche nous avons remarqué que le PASCAL VOC 2012 [1] et 2007 sont des références en la matière. Il propose des challenges de détection et de segmentation d'image basés sur le principe du "ground truth". Chaque image est segmentée et labellée "manuellement" par un humain. Le but de la machine est donc de se rapprocher le plus possible de ce "ground truth". C'est un dataset intéressant notamment parce qu'il est bien adapté à l'apprentissage deep-learning mais aussi parce qu'il possède une très grande collection d'objet, 27450 objets pour la détection (sur 11450 images) et 6929 pour la segmentation (2913 images), le tout sur plus de 2Go.

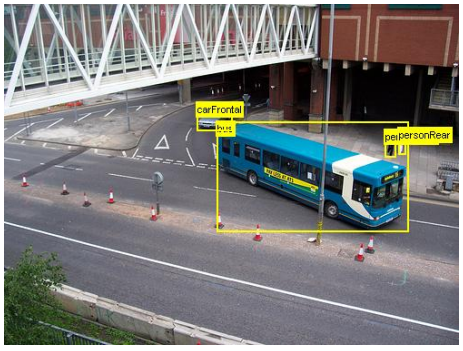


Figure 2: La detection

Figure 3: La segmentation

. Maintenant que nous avons défini ce qu'est le deep-learning et que l'on possédons un dataset commun pour l'entraînement et le test, nous allons présenter au travers d'articles les principales technologies à la pointe aujourd'hui.

2.3. Un état de l'art des technologies

2.3.1. Fully convolutionnal network

. Une première approche de mars 2015 [2] est simplement un enchaînement de couches de convolution pour segmenter l'image en région de pixel. Le but n'est pas comme pour les couches de "fully connected layer" de prédire ce que l'image représente le plus mais bien de fournir une "heat map" des objets reconnus. L'entraînement se fait comme pour n'importe quel réseau neuronal convolutif. (Comme montré sur la figure 4)

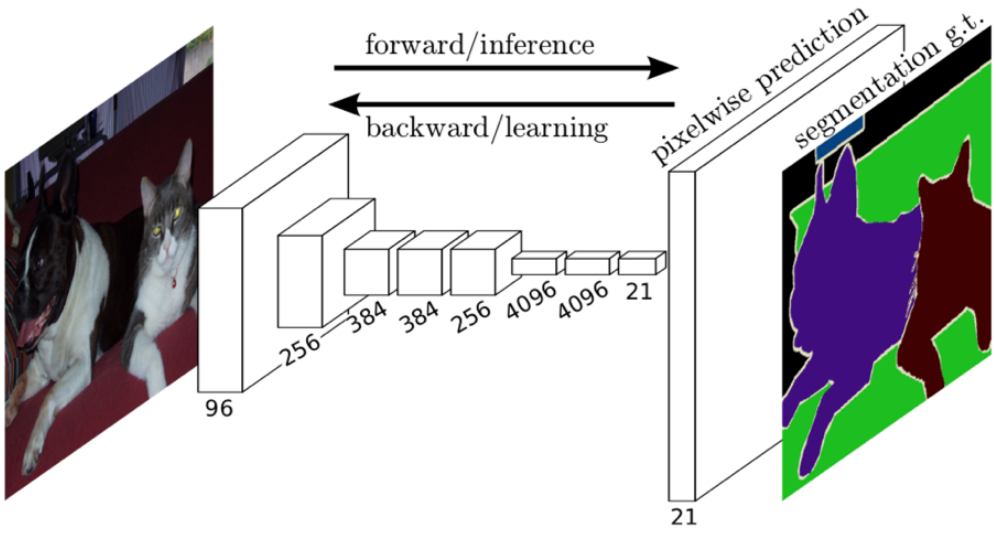


Figure 4: Fully convolutionnal network

2.3.2. Deep Residual Learning for image recognition

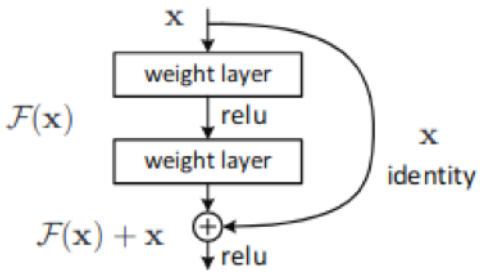


Figure 5: Deep Residual Learning

. Cette méthode [3] permet de palier à la dégradation des données lorsque l'on empile une grande quantité de couches. Au lieu d'espérer que chaque couche empilée corresponde directement à un mappage sous-jacent souhaité (noté H(x)), on laisse explicitement ces couches s'adapter à un mappage résiduel. Les couches non-linéaires empilées s'adaptent à une autre application F(x) tel que F(x)=H(x)+x. Les connexions peuvent se faire entre couches jointes ou pas.

2.3.3. Pyramid Scene Parsing

. Cette méthode [4] a pour vocation de restreindre la perte d'information entre les différentes régions. Elle utilise ResNet (Residual Network) pour obtenir une première carte des caractéristiques. Puis elle crée une pyramide. Les étages de celle-ci permettent de séparer les différentes caractéristiques en multiples régions et représentations afin d'obtenir de nouvelles cartes de caractéristiques. Elles sont ensuite toutes fusionnées.

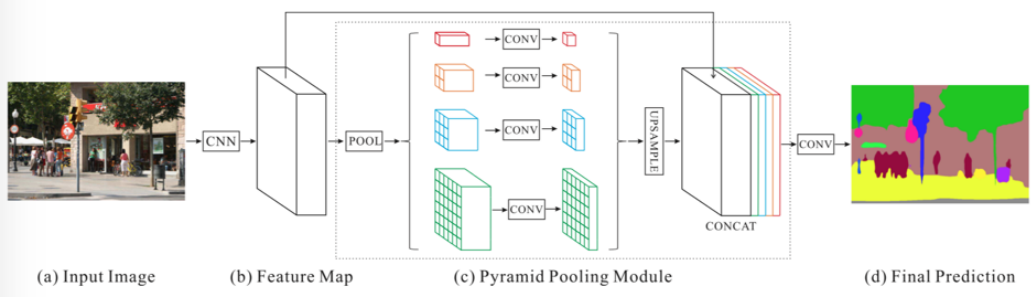


Figure 6: Pyramid Scene Parsing

2.3.4. SegNet

. SegNet [5] est une approche qui fonctionne plutôt bien puisque de nombreux modèles sont basés sur celui-ci. Le principe derrière est très simple, appliquer la méthode bien connue en deep-learning de l'encodeur-décodeur à une méthode efficace dans le domaine de la vision la convolution.

. Dans un premier temps, on lui apprend des features que l'on "encode" à l'aide de couche de convolution avec des pooling (pour réduire la map de représentation). Ensuite on upscale et fait des convolutions pour détecter les groupes de pixel et donc segmenter l'image. Ensuite l'on peut coupler ce modèle pour faire de la segmentation et détection sémantique.

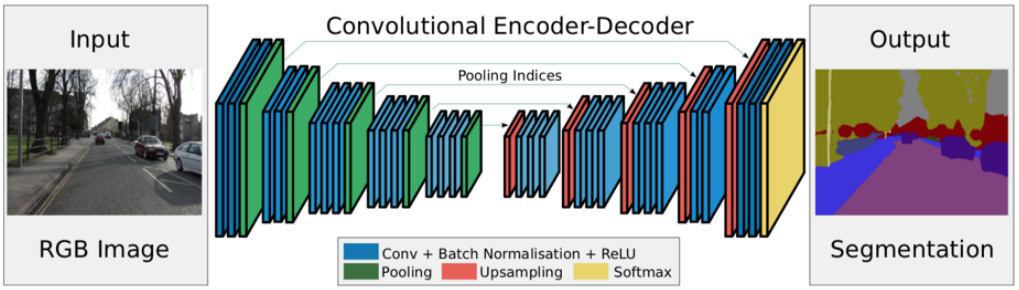


Figure 7: SegNet par encodeur/decodeir à convolution

2.3.5. DeepLayer Cascade

. Le layer cascade [6] hérite des avantages de modèles en cascade ń conventionnels ż. Il possède plusieurs couches et entraîne un paramètre par couche. Le Layer Cascade considère les différentes couches du réseau comme ń différentes étapes ż, les dernières couches ne servant qu'à affiner le travail des couches précédentes dont les résultats sont déjà considérés comme digne de confiance. Cela permet d'éviter les faux positifs. Le Layer cascade permet donc d'augmenter le taux de détection des échantillons en se basant sur des modèles ayant déjà fait leur preuve mais aussi de réduire de manière significative le temps de calcul.

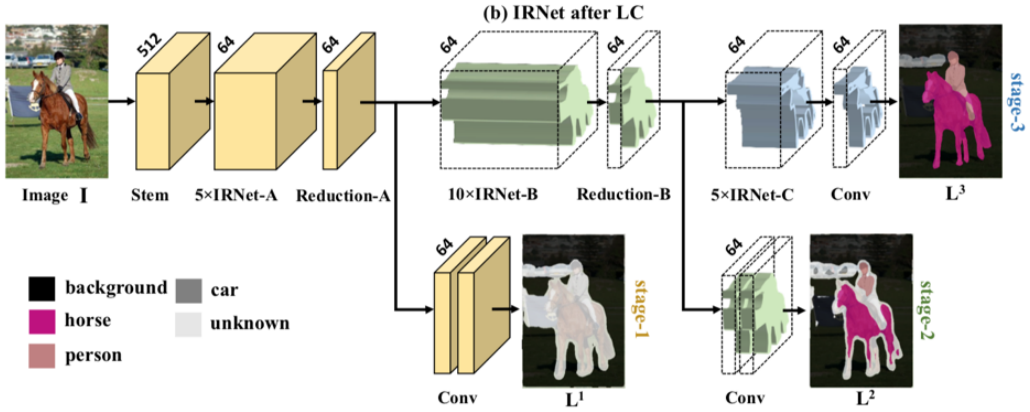


Figure 8: DeepLayer Cascade

2.4. Les derniers résultats

. Dans le dernier papier publié sur ce thème, un tableau (voir ci-dessous) qui montre la performance au mIOU (mean Intersection over Union) sur la segmentation d'image et il en ressort que le DeepLavv3+ Xception est le plus efficace au 8 mars 2018. Nous allons donc étudier son fonctionnement détaillé au travers de l'article qui le décrit [7]. Mais aussi de l'article de la première version de deeplab [8] . Nous le mettrons également en parallèle avec un article d'un modèle fondateur dans le milieu, Fast RCNN [9] de 2015 pour montrer l'évolution fulgurante dans ce secteur en 3 ans.

Method	**mIOU**
Deep Layer Cascade (LC) [42]	82.7
TuSimple [75]	83.1
Large_Kernel_Matters [57]	83.6
Multipath-RefineNet [43]	84.2
ResNet-38_MS_COCO [77]	84.9
PSPNet [81]	85.4
IDW-CNN [73]	86.3
CASIA_IVA_SDN [20]	86.6
DIS [50]	86.8
DeepLabv3 [10]	85.7
DeepLabv3-JFT [10]	86.9
DeepLabv3+ (Xception)	87.8
DeepLabv3+ (Xception-JFT)	89.0

Figure 9: Résultats au IoU

3. Analyse

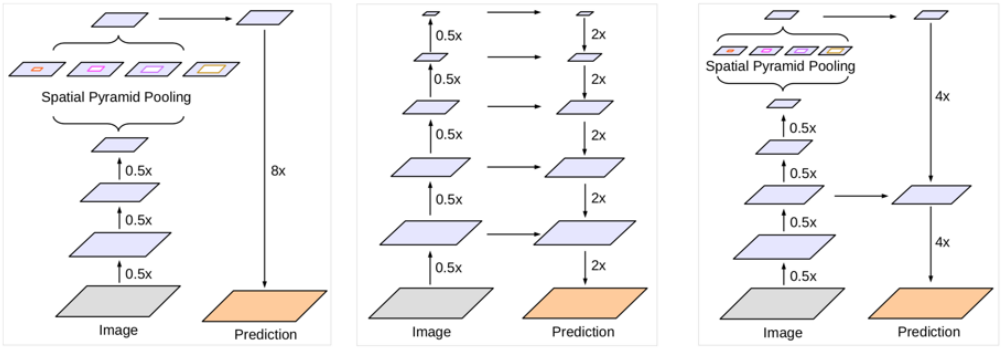
3.1. Encoder-decoder with atrous separable convolution for semantic image segmentation

. L'article (réponse possible à notre problématique) que nous allons analyser a pour titre "Encoder-decoder with atrous separable convolution for semantic image segmentation" qui propose le modèle DeeplabV3+. Mais nous allons aussi faire référence à la première version de deeplab qui date de juin 2017 ainsi qu'au Fast RCNN, un des modèles les plus connus dans le domaine.

. L'article est de L-C Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, chercheur chez Google. Google est l'une des entreprises qui s'est la plus investie dans l'intelligence artificielle ces dernières années, aidée par d'important moyen financier et surtout par l'incroyable masse de données qu'elle possède.

. Pour capturer les informations contextuelles à différentes échelles, on applique beaucoup de ASPP (Asynchronous Spatial Pyramid Pooling) en parallèle. Bien que la plupart des informations soient encodées dans la dernière map de feature, les informations détaillées par rapport aux contours sont perdues à cause des pooling ou de la convolution dans le corps du réseau.

. En appliquant la convolution à trous, on a une résolution en sortie plus petite que la résolution d'entrée. Il existe des structures de encoder/decoder permettant de calculer plus rapidement et de récupérer graduellement les formes. En combinant les 2 méthodes on obtient alors un meilleur contrôle des ressources utilisées, une meilleure détection et une bonne détection des contours. D'où la structure ci-dessous



(a) Spatial Pyramid Pooling (b) Encoder-Decoder (c) Encoder-Decoder with Atrous Conv.

Figure 10: Structure

. Les principales améliorations qu'apportent cet article, notamment par rapport au premier de mai 2017, sont un nouveau couple encodeur/décodeur avec un encodeur puissant, un décodeur simple, et une adaptation du modèle Xception qui est un modèle de classification des plus efficace actuellement (meilleur que VGG, Resnet ou bien inception).

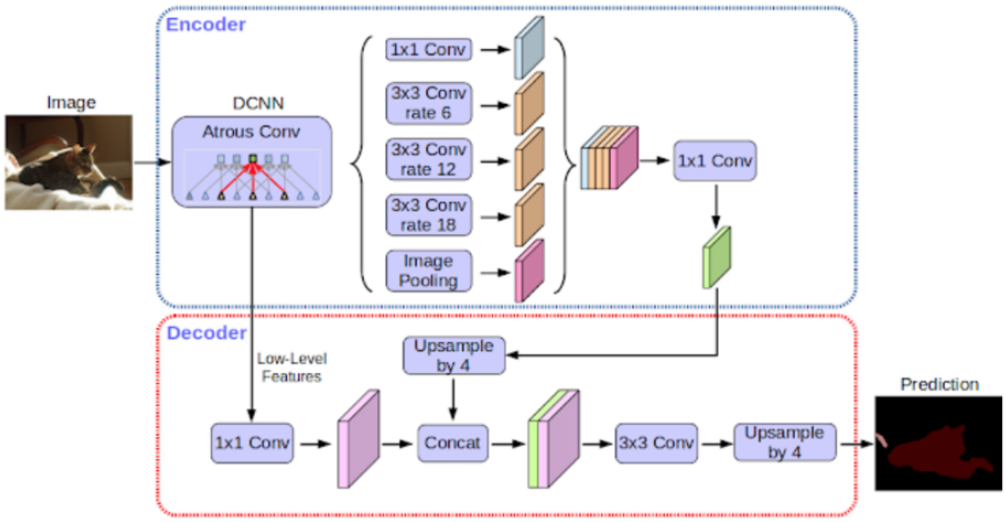


Figure 11: Architecture complète du modèle

. Les technologies qui sous-tendent Deeplabv3 sont :

- Le "Spatial Pyramid pooling" qui vient du PSPNet va faire de convolutions sur différentes échelles et donc permettre de faire de la détection d'objet sur du multi-échelle.
- Les "encodeurs/decodeurs" font une représentation à l'aide de convolution et vont reconstituer la sortie à partir de la map de feature.
- La méthode de "Depthwise separable convolution" qui permet de faire des groupes de convolution et réduire le nombre de calcul, c'est la méthode utilisé par le modàle Xception.

. Une méthode très intéressante développée dans les modèles deeplab est la "atrous convolution", ou convolution à trous. La solution initialement créée utilise le max-pooling et striding cependant cette méthode provoque une baisse de résolution de l'image. Pour pallier ce défaut on utilise des couches de déconvolution nécessitant beaucoup de calculs et une grosse quantité de mémoire. Atrous convolution vient de "l'algorithme à trous" qui introduit des trous dans le filtre (le cas classique de la convolution a pour valeur r = 1). Ici on va mettre différents écarts [Figure 13]

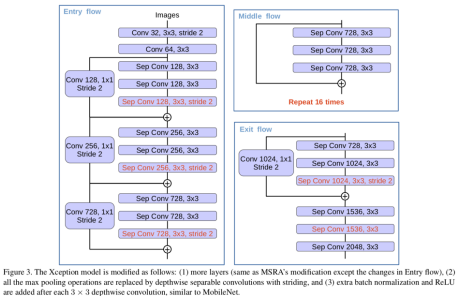


Figure 3. The Xception model is modified as follows: (1) more layers (same as MSRA's modification except the changes in Entry flow), (2) all the max pooling operations are replaced by depthwise separable convolutions with striding, and (3) extra batch normalization and ReLU are added after each 3×3 depthwise convolution, similar to MobileNet.

Figure 12: Architecture de Xception

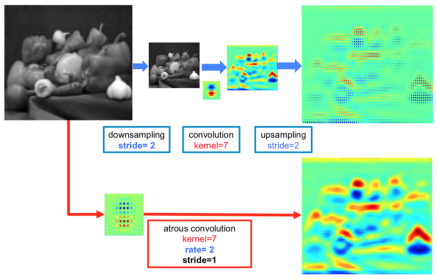


Figure 13: différence de fonctionnement entre une convolution avec downsampling/upsampling et la convolution à trous

. Les évaluations expérimentales ont été nécessaires pour déterminer quel décodeur ou quel réseau de classification sont les meilleurs. En dessous nous observons les résultats IoU (Intersect over Union) en fonction du nombre de pixels de la trimap (une image sur laquelle on indique l'arrière-plan et l'objet). On voit que les méthodes "classiques" d'upscaling (bilinear upscaling) sont moins efficaces que la méthode avec le décodeur de deeplab. Le modèle pour la classification le plus efficace est Xception modifié pour ce cas.

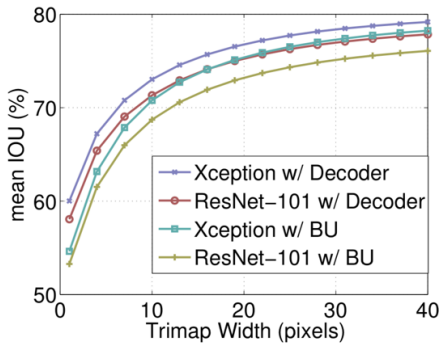


Figure 4. mIOU as a function of the trimap band width around the object boundaries when employing *train output stride* = *eval output stride* = 16. **BU**: Bilinear upsampling.

Figure 14: Evaluation du meilleur modele

(a) Image (b) w/ BU (c) w/ Decoder

Figure 15: Différence de segmentation entre le BU et le decoder

. DeepLabv3 est en grande partie basé sur DeepLabV1 de mi 2017. Il introduit notamment la convolution à trous, mais aussi la prédiction structurée

avec une couche "fully-connected conditionnal random field". Méthode qui n'a donc pas été gardée dans Deeplabv3+ se servant de Xception et d'un système de encodeur/décodeur. Ainsi passe-t-on d'un résultat de 79.7 au mIoU à 89 en moins de 1 an !

. L'écart est encore plus impressionnant avec l'approche Fast-RCNN de 2016, une des premières approches reconnues dans le domaine.

3.2. FAST R-CNN

. L'article (2015) sur lequel s'appuie cette étude a été rédiger par Ross Girshick, un chercheur de l'entreprise Microsoft.

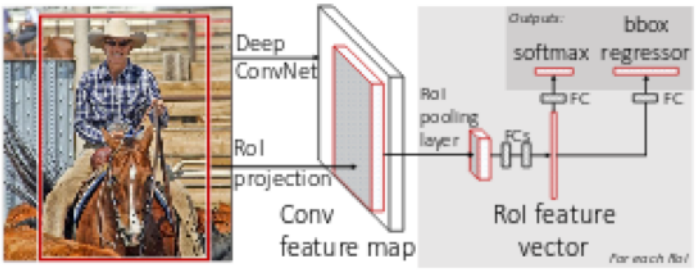


Figure 16: Fonctionnement de Fast R-CNN

. Le Fast-RCNN s'effectue en 3 étapes. Tout d'abord on utilise un réseau convolutif et du pooling pour créer une carte des caractéristiques que l'on divise ensuite en régions d'intérêt selon les vecteurs de caractéristiques rassemblés en couches fully-connected. Puis grâce à ces informations, on estime l'objet représenté sur l'image. On affine ce choix par la rétropropagation et la descente du gradient.

. Mais cette dernière opération est très coûteuse en temps c'est pourquoi elle est très souvent associée à du multitasking. A chaque étape de travail sur les régions d'intérêt, on calcule la probabilité de l'objet d'appartenir aux différentes classes d'objets mises en entrée et on élimine au fur et à mesure les options absurdes. En effet cette méthode est plus rapide car pour la classification, le temps dédié à créer le réseau fully connected est négligeable devant celui dédié aux couches convolutives.

. Le Fast-RCNN peut encore être accéléré grâce à des méthodes de compression quand le nombre de régions d'intérêt est trop important (cela consiste en la division de la couche fully connected).

. Le Fast-RCNN a un résultat de 66 au mIoU. Cette performance est nettement inférieure au DeepLabv1 et DeepLabv3 étudiés plus haut. Cependant il reste très utilisé dans les algorithmes de détection d'aujourd'hui.

3.3. Les méthodes les plus prometteuses pour l'avenir

. Problème de l'IA n'est plus d'améliorer les détections et les segmentations car avec ces résultats l'IA est aujourd'hui meilleure que l'humain. Mais dorénavant, le problème qui se pose est la puissance de calcul nécessaire (et donc le temps de traitement de l'image) et surtout l'intégration dans d'autres systèmes plus complexe (comme les voitures autonomes). Le tout en se devant de garantir des résultats toujours bons (malgré le fait que ceux sont des systèmes non-déterministes).

. On pense aujourd'hui que les méthodes les plus prometteuses sont celles qui pourront combiner plusieurs blocs pour tirer un meilleur parti de chacun (comme avec l'exemple de DeepLabv3+). Ainsi que celles qui utilisent des modèles dont la fiabilité n'est plus à démontrer, garantissant une certaine confiance.

3.4. Critiques de ces méthodes

. Une grande force de ces approches DeepLearning pour la vision c'est qu'elles sont encore jeunes mais font déjà leurs preuves en affichant de très bons résultats. Elles promettent un avenir passionnant avec de meilleurs résultats. Aujourd'hui les méthodes restent encore très diversifiées et permettent une exploration d'autant plus rapide des différents aspects dans ce domaine.

. Malgré tout, cette jeunesse reste une faiblesse car les résultats restent perfectibles et les applications peu nombreuses, même s'il y en a de plus en plus. Nous trouvons que la recherche dans ce secteur manque de rigueur, avec des modèles qui ont l'air "rafistolés". C'est-à-dire que l'on conçoit des blocs, pour les mettre ensemble, on entraîne l'algorithme et on le teste. Certains modèles sont construits à partir ou grâce à des modèles plus anciens, en essayant de combler les failles des précédents. Cela reste à l'opposé de ce que pourraient faire des méthodes plus formelles et déterministes.

4. Conclusion

. L'intérêt de ces recherches est indéniablement dans l'analyse de l'image et a fortiori dans la vision par ordinateur. Depuis leur développement fulgurant, les voitures autonomes ont besoin de voir leur environnement pour le comprendre et faire des choix. Cette robotique plus intelligente va devenir omniprésente dans nos vies.

. Nous ne sommes qu'au début de la révolution que prépare la vision par ordinateur et plus généralement de l'intelligence artificielle deep-learning. On peut supposer que la machine sera dans les prochaines années bien plus efficace que l'humain dans l'analyse des scènes et situations. Elle permettra de mieux comprendre certaines choses qui ne nous sont pas instinctives.

References

[1] PASCAL VOC 20012, *http://host.robots.ox.ac.uk/pascal/VOC/voc2012/*

[2] Fully convolutionnal network, *https://arxiv.org/pdf/1411.4038.pdf*

[3] Deep Residual Learning, *https://arxiv.org/pdf/1512.03385.pdf*

[4] Pyramid Scene Parsing, *https://arxiv.org/pdf/1612.01105.pdf*

[5] SegNet, *https://arxiv.org/pdf/1511.00561.pdf*

[6] DeepLayer Cascade, *https://arxiv.org/pdf/1704.01344.pdf*

[7] DeepLab V3+, *https://arxiv.org/pdf/1802.02611.pdf*

[8] DeepLab *https://arxiv.org/pdf/1606.00915.pdf*

[9] Faster R-CNN, *https://arxiv.org/pdf/1504.08083.pdf*