Réseaux entièrement convolutifs pour la segmentation sémantique

Jonathan Long* Evan Shelhamer* Trevor Darrell
UC Berkeley
{jonlong, shelhamer, trevor} @ cs.berkeley.edu

Abstrait

*Les réseaux convolutifs sont des modèles visuels puissants qui génèrent des hiérarchies de fonctionnalités. Nous montrons que convolu-
Les réseaux nationaux par eux-mêmes, entraînés de bout en bout, pixels à pixels, dépassent l'état de l'art en matière de segmentation sémantique. Notre idée clé est de construire des réseaux «entièrement convolutifs» qui prennent des entrées de taille arbitraire et produisent des sorties de taille correspondante avec une inférence et un apprentissage ef fi caces. Nous dé fi nissons et détaillons l'espace des réseaux entièrement convolutifs, expliquons leur application à des tâches de prédiction spatialement denses et établissons des connexions avec des modèles antérieurs. Nous adaptons les réseaux de classi fi cation contemporains (AlexNet [22], le réseau VGG [34] et GoogLeNet [35]) dans des réseaux entièrement convolutifs et transférer leurs représentations apprises par un réglage fin [5] à la tâche de segmentation. Nous définissons ensuite une architecture de saut qui combine des informations sémantiques d'une couche profonde et grossière avec des informations d'apparence d'une couche fine et superficielle pour produire des segmentations précises et détaillées. Notre réseau entièrement convolutif réalise une segmentation de pointe de PASCAL VOC (amélioration relative de 20% à 62,2% UI moyenne en 2012), NYUDv2 et SIFT Flow, tandis que l'inférence prend moins d'un cinquième de seconde pour une image typique.*

1. Introduction

Les réseaux convolutifs font progresser la reconnaissance. Les convnets ne s'améliorent pas seulement pour la classification de l'image entière [22, 34, 35], mais aussi progresser sur les tâches locales avec des résultats structurés. Celles-ci incluent des avancées dans la détection des objets de la boîte englobante [32, 12, 19], prédiction de pièces et de points clés [42, 26] et la correspondance locale [26, dix].

La prochaine étape naturelle dans la progression de l'inférence grossière à l'inférence fine consiste à faire une prédiction à chaque pixel. Des approches antérieures ont utilisé des convnets pour la segmentation sémantique [30, 3, 9, 31, 17, 15, 11], dans lequel chaque pixel est étiqueté avec la classe de son objet ou de sa région englobante, mais avec des inconvénients que ce travail aborde.

*Les auteurs ont contribué à parts égales

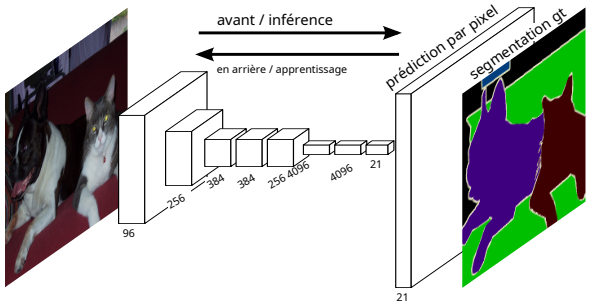

Figure 1. Les réseaux entièrement convolutifs peuvent ef fi cacement apprendre à faire des prédictions denses pour les tâches par pixel comme la segmentation sémantique.

Nous montrons qu'un réseau entièrement convolutif (FCN) formé de bout en bout, pixels à pixels sur une segmentation sémantique, dépasse l'état de l'art sans autre machinerie. À notre connaissance, il s'agit du premier travail pour former les FCN de bout en bout (1) pour la prédiction pixel par pixel et (2) à partir d'un pré-apprentissage supervisé. Les versions entièrement convolutives des réseaux existants prédisent des sorties denses à partir d'entrées de taille arbitraire. L'apprentissage et l'inférence sont exécutés image entière à la fois par un calcul et une rétropropagation dense par anticipation. Les couches de suréchantillonnage en réseau permettent une prédiction et un apprentissage au niveau des pixels dans des réseaux avec un regroupement sous-échantillonné.

Cette méthode est ef fi cace, à la fois asymptotiquement et de manière absolue, et exclut la nécessité de complications dans d'autres travaux. La formation patchwise est courante [30, 3, 9, 31, 11], mais n'a pas l'ef fi cacité d'une formation entièrement convolutive. Notre approche n'utilise pas les complications de pré- et post-traitement, y compris les superpixels [9, 17], les propositions [17, 15], ou un raffinement post-hoc par des champs aléatoires ou des classificateurs locaux [9, 17]. Notre modèle transfère le succès récent de la classification [22, 34, 35] à la prédiction dense en réinterprétant les réseaux de classification comme étant entièrement convolutifs et ajustés à partir de leurs représentations apprises. En revanche, les travaux précédents ont appliqué de petits réseaux sans pré-formation supervisée [9, 31, 30].

La segmentation sémantique est confrontée à une tension inhérente entre la sémantique et la localisation: l'information globale résout quoi tandis que l'information locale résout où. Les hiérarchies de fonctionnalités approfondies codent l'emplacement et la sémantique de manière non linéaire

pyramide locale à globale. Nous définissons une architecture de saut pour tirer parti de ce spectre de fonctionnalités qui combine des informations sémantiques profondes, grossières et des informations d'apparence superficielles et fines dans la section4.2 (voir la figure 3).

Dans la section suivante, nous passons en revue les travaux connexes sur les réseaux de classification profonde, les FCN et les approches récentes de la segmentation sémantique utilisant des convnets. Les sections suivantes expliquent la conception FCN et les compromis de prédiction dense, présentent notre architecture avec suréchantillonnage en réseau et combinaisons multicouches, et décrivent notre cadre expérimental. Enfin, nous démontrons des résultats de pointe sur PASCAL VOC 2011-2, NYUDv2 et SIFT Flow.

2. Travaux connexes

Notre approche s'appuie sur les récents succès des réseaux profonds pour la classification d'images [22, 34, 35] et transfert d'apprentissage [5, 41]. Le transfert a été démontré pour la première fois sur diverses tâches de reconnaissance visuelle [5, 41], puis sur la détection, et sur la segmentation d'instance et sémantique dans les modèles hybrides proposition-classi fi er [12, 17, 15]. Nous restructurons et affinons maintenant les réseaux de classi fi cation pour une prédiction directe et dense de la segmentation sémantique. Nous cartographions l'espace des FCN et situons les modèles antérieurs, à la fois historiques et récents, dans ce cadre.

Réseaux entièrement convolutifs À notre connaissance, le l'idée d'étendre un convnet à des entrées de taille arbitraire est apparue pour la première fois à Matan *et coll. [*28], qui a étendu le LeNet classique [23] pour reconnaître des chaînes de chiffres. Parce que leur réseau était limité à des chaînes d'entrée unidimensionnelles, Matan*et coll.* utilisé le décodage Viterbi pour obtenir leurs sorties. Wolf et Platt [40] étendre les sorties convnet à des cartes bidimensionnelles des scores de détection pour les quatre coins des blocs d'adresses postales. Ces deux travaux historiques font des inférences et des apprentissages entièrement convolutifs pour la détection. Ning*et coll. [*30] dé fi nissez un convnet pour une segmentation multiclasse grossière de *C. elegans* tissus avec inférence entièrement convolutive.

Le calcul entièrement convolutif a également été exploité à l'ère actuelle des réseaux à plusieurs couches. Détection de fenêtre coulissante par Sermanet*et coll. [*32], segmentation sémantique par Pinheiro et Collobert [31], et la restauration d'image par Eigen *et coll. [*6] font une inférence entièrement convolutive. La formation entièrement continue est rare, mais utilisée efficacement par Tompson *et coll. [*38] pour apprendre un détecteur de pièces de bout en bout et un modèle spatial pour l'estimation de pose, bien qu'ils n'exposent pas ou n'analysent pas cette méthode.

Alternativement, il *et coll. [*19] supprime la partie non convolutive des réseaux de classification pour créer un extracteur de caractéristiques. Ils combinent des propositions et un regroupement de pyramides spatiales pour produire une caractéristique localisée de longueur fixe pour la classification. Bien que rapide et efficace, ce modèle hybride ne peut pas être appris de bout en bout.

Prédiction dense avec convnets Plusieurs travaux récents ont appliqué des convnets à des problèmes de prédiction denses, y compris la segmentation sémantique par Ning *et coll. [*30], Farabet *et coll.* [9], et Pinheiro et Collobert [31]; prédiction des limites pour la microscopie électronique par Ciresan*et coll. [*3] et pour les images naturelles par un modèle hybride convnet / plus proche voisin de Ganin et Lempitsky [11]; et restauration d'image et estimation de la profondeur par Eigen*et coll. [*6, 7]. Les éléments communs de ces approches comprennent

- petits modèles limitant la capacité et les champs réceptifs;
- formation patchwise [30, 3, 9, 31, 11];
- post-traitement par projection superpixel, régularisation de champ aléatoire, filtrage ou classi fi cation locale [9, 3, 11];
- décalage d'entrée et entrelacement de sortie pour une sortie dense [32, 31, 11];
- traitement pyramidal multi-échelles [9, 31, 11];
- saturant Tanh non-linéarités [9, 6, 31]; et
- ensembles [3, 11],

alors que notre méthode se passe de cette machinerie. Cependant, nous étudions la formation patchwise3.4 et sortie dense «shift-and-stitch» 3.2 du point de vue des FCN. Nous discutons également du suréchantillonnage en réseau3,3, dont la prédiction entièrement connectée par Eigen *et coll. [*7] est un cas particulier.

Contrairement à ces méthodes existantes, nous adaptons et étendons les architectures de classi fi cation en profondeur, en utilisant la classi fi cation d'image comme pré-apprentissage supervisé, et nous affinons entièrement par convolution pour apprendre simplement et ef fi cacement à partir d'entrées d'images entières et de vérités globales d'image.

Hariharan *et coll. [*17] et Gupta *et coll. [*15] adaptent également les réseaux de classification profonde à la segmentation sémantique, mais le font dans les modèles hybrides proposition-classi fi cation. Ces approches affinent un système R-CNN [12] en échantillonnant des boîtes englobantes et / ou des propositions de régions pour la détection, la segmentation sémantique et la segmentation d'instances. Aucune des deux méthodes n'est apprise de bout en bout. Ils obtiennent des résultats de segmentation de pointe sur PASCAL VOC et NYUDv2 respectivement, nous comparons donc directement notre FCN autonome de bout en bout à leurs résultats de segmentation sémantique dans la section5.

Nous fusionnons les entités à travers les couches pour définir une représentation locale à globale non linéaire que nous ajustons de bout en bout. Dans le travail contemporain Hariharan*et coll. [*18] utilisent également plusieurs couches dans leur modèle hybride pour la segmentation sémantique.

3. Réseaux entièrement convolutifs

Chaque couche de données dans un convnet est un tableau tridimensionnel de taille $h \times w \times ré$, où h et w sont des dimensions spatiales, et $ré$ est la dimension de la fonction ou du canal. Le premier calque est l'image, avec une taille de pixel $h \times w$, et $ré$ canaux de couleur. Les emplacements dans les couches supérieures correspondent aux emplacements de l'image auxquels ils sont connectés au chemin, qui sont appelés leurs *champs réceptifs.*

Les convnets sont construits sur l'invariance de la traduction. Leurs composants de base (fonctions de convolution, de mise en commun et d'activation) opèrent sur les régions d'entrée locales et ne dépendent que de *relatif* coordonnées spatiales. L'écriture X_{ij} pour le vecteur de données à l'emplacement (i, j) dans une couche particulière, et y_{ij} pour la suite

couche, ces fonctions calculent les sorties y_{ij} par

$$y_{ij} = F_{ks}(\{x_{si+\delta i,\, sj+\delta j}\}_{0 \le \delta i, \delta j \le k})$$

où k s'appelle la taille du noyau, s est la foulée ou le sous-groupe facteur de pling, et F_{ks} détermine le type de couche: une multiplication matricielle pour la convolution ou la mise en commun moyenne, une max pour le pooling max, ou une non-linéarité élémentaire pour une fonction d'activation, et ainsi de suite pour d'autres types de couches.

Cette forme fonctionnelle est maintenue en composition, la taille du noyau et la foulée obéissant à la règle de transformation

$$F_{ks} \circ g_{k's'} = (F \circ g)_{k'+(k-1)s',\, ss'}$$

Alors qu'un réseau profond général calcule une fonction non linéaire générale, un réseau avec uniquement des couches de cette forme calcule une fonction non linéaire *filtre,* que nous appelons un *fi ltre profond* ou alors *réseau entièrement convolutif.* Un FCN fonctionne naturellement sur une entrée de n'importe quelle taille et produit une sortie de dimensions spatiales correspondantes (éventuellement rééchantillonnées).

Une fonction de perte à valeur réelle composée avec un FCN détermine une tâche. Si la fonction de perte est une somme sur l'espace dimensions de la couche fi nale, $\ell(X; \theta) = \sum_{ij} \ell'(x_{ij}; \theta)$, son gradient sera une somme sur les gradients de chacune de ses composantes spatiales. Ainsi, la descente de gradient stochastique sur ℓ calculée sur des images entières sera la même que la descente de gradient stochastique sur ℓ', en prenant tous les champs récepteurs de la couche fi nale comme un mini-lot.

Lorsque ces champs réceptifs se chevauchent de manière signi fi cative, les deux calculs *et* La rétropropagation est beaucoup plus ef fi cace lorsqu'elle est calculée couche par couche sur une image entière plutôt que indépendamment patch-par-patch.

Nous expliquons ensuite comment convertir les réseaux de classification en réseaux entièrement convolutifs qui produisent des cartes de sortie grossières. Pour la prédiction par pixel, nous devons connecter ces sorties grossières aux pixels. Section3.2 décrit une astuce, une numérisation rapide [13], introduit à cet effet. Nous obtenons un aperçu de cette astuce en la réinterprétant comme une modification de réseau équivalente. Comme alternative ef fi cace et efficace, nous introduisons des couches de déconvolution pour le suréchantillonnage dans la section3,3. Dans la section 3.4 nous considérons la formation par échantillonnage patchwise, et donnons des preuves dans la section 4.3 que l'ensemble de notre formation à l'image est plus rapide et tout aussi efficace.

3.1. Adaptation des classificateurs pour une prédiction dense

Réseaux de reconnaissance typiques, y compris LeNet [23], AlexNet [22], et ses successeurs plus profonds [34, 35], prennent ostensiblement des intrants de taille fi xe et produisent des extrants non spatiaux. Les couches entièrement connectées de ces réseaux ont des dimensions fixes et rejettent les coordonnées spatiales. Cependant, ces couches entièrement connectées peuvent également être considérées comme des convolutions avec des noyaux qui couvrent l'ensemble de leurs régions d'entrée. Cela les transforme en réseaux entièrement convolutifs qui prennent des entrées de n'importe quelle taille et des cartes de classi fi cation en sortie. Cette transformation est illustrée dans la figure2.

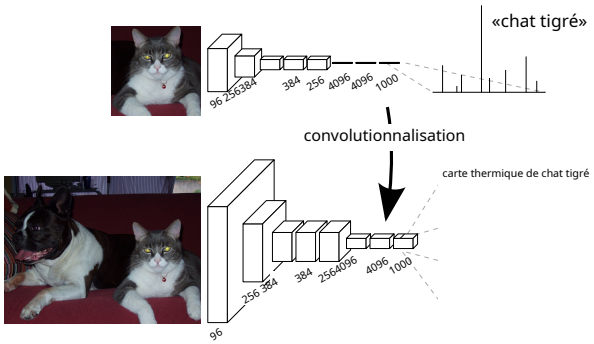

Figure 2. La transformation de couches entièrement connectées en couches de convolution permet à un réseau de classification de produire une carte thermique. Ajout de couches et d'une perte spatiale (comme dans la figure1) produit une machine ef fi cace pour un apprentissage dense de bout en bout.

En outre, alors que les cartes résultantes sont équivalentes à l'évaluation du réseau d'origine sur des patchs d'entrée particuliers, le calcul est fortement amorti sur les régions de chevauchement de ces patchs. Par exemple, alors qu'AlexNet prend 1.2 ms (sur un GPU typique) pour déduire les scores de classi fi cation d'un 227×227 image, le réseau entièrement convolutif prend 22 ms pour produire un dix×dix grille de sorties d'un 500×500 image, qui est plus que 5 fois plus rapide que l'approche naïve[1].

Les cartes de sortie spatiale de ces modèles convolutalisés en font un choix naturel pour des problèmes denses comme la segmentation sémantique. Avec la vérité terrain disponible à chaque cellule de sortie, les passes avant et arrière sont simples et toutes deux tirent parti de l'ef fi cacité de calcul inhérente (et de l'optimisation agressive) de la convolution. Les temps de retour correspondants pour l'exemple AlexNet sont2,4 ms pour une seule image et 37 ms pour un tout convolutif dix × dix carte de sortie, résultant en une accélération similaire à celle de la passe avant.

Alors que notre réinterprétation des réseaux de classification en tant que réseaux entièrement convolutifs donne des cartes de sortie pour des entrées de toute taille, les dimensions de sortie sont généralement réduites par le sous-échantillonnage. Le sous-échantillon des réseaux de classification permet de garder les fi ltres petits et les besoins de calcul raisonnables. Cela grossit la sortie d'une version entièrement convolutive de ces réseaux, la réduisant de la taille de l'entrée d'un facteur égal au pas de pixel des champs récepteurs des unités de sortie.

3.2. Shift-and-stitch est une raréfaction du filtre

Des prédictions denses peuvent être obtenues à partir de sorties grossières en assemblant la sortie de versions décalées de l'entrée. Si la sortie est sous-échantillonnée d'un facteur de$f$, décale l'entrée x pixels à droite et y pixels vers le bas, une fois pour chaque (x, y) st $0 \le x, y < f$. Traitez chacun de ces f^2 entrées, et entrelacez les sorties de sorte que les prédictions correspondent aux pixels au *centres* de leurs champs réceptifs.

[1] En supposant un traitement par lots ef fi cace des entrées d'image unique. Les scores de classi fi cation pour une seule image prennent à eux seuls 5,4 ms à produire, ce qui est presque 25 fois plus lent que la version entièrement convolutive.

Bien qu'effectuer cette transformation augmente naïvement le coût d'un facteur de F^2, il existe une astuce bien connue pour produire ef fi cacement des résultats identiques [13, 32] connu de la communauté des ondelettes sous le nom d'algorithme à trous [27]. Considérons une couche (convolution ou pooling) avec foulée d'entrée s, et une couche de convolution ultérieure avec des poids de filtre f_{ij} (élimination des cotes de fonction non pertinentes). Réglage de la foulée d'entrée de la couche inférieure sur1 suréchantillonne sa sortie par un facteur de s. Cependant, la convolution du fi ltre original avec la sortie suréchantillonnée ne produit pas le même résultat que le décalage et le point, car le fi ltre original ne voit qu'un rééchantillonnage. portion {de son entrée (maintenant suréchantillonnée). Reproduire l'astuce, raréfiez le fi ltre en l'agrandissant comme

$$f'_{ij} = \begin{cases} f_{i/s,\,j/s} & \text{si } s \text{ divise les deux } je \text{ et } j; \\ 0 & \text{autrement,} \end{cases}$$

(avec je et j base zéro). Reproduire l'intégralité de la sortie nette L'astuce consiste à répéter cet agrandissement du filtre couche par couche jusqu'à ce que tout le sous-échantillonnage soit supprimé. (En pratique, cela peut être fait ef fi cacement en traitant les versions sous-échantillonnées de l'entrée suréchantillonnée.)

Diminuer le sous-échantillonnage au sein d'un réseau est un compromis: les fi ltres voient des informations plus fines, mais ont des champs réceptifs plus petits et prennent plus de temps à calculer. L'astuce shift-and-stitch est un autre type de compromis: la sortie est plus dense sans diminuer les tailles de champ de réception des fi ltres, mais il est interdit aux fi ltres d'accéder aux informations à une échelle inférieure à leur conception d'origine.

Bien que nous ayons fait des expériences préliminaires avec cette astuce, nous ne l'utilisons pas dans notre modèle. Nous trouvons que l'apprentissage par suréchantillonnage, comme décrit dans la section suivante, est plus efficace et plus ef fi cace, en particulier lorsqu'il est combiné avec la fusion de couches de sauts décrite plus loin.

3.3. Le suréchantillonnage est une convolution inversée

L'interpolation est une autre façon de connecter des sorties grossières à des pixels denses. Par exemple, une simple interpolation bilinéaire calcule chaque sortie y_{ij} à partir des quatre entrées les plus proches par une carte linéaire qui ne dépend que des positions relatives des cellules d'entrée et de sortie.

Dans un sens, suréchantillonnage avec facteur F est la convolution avec un *fractionnaire* pas d'entrée de $1/F$. Tant que F est intégral, une manière naturelle de suréchantillonner est donc *convolution vers l'arrière* (appelé quelques fois *déconvolution)* avec un *production* enjambée de F. Une telle opération est simple à mettre en œuvre, car elle inverse simplement les passes avant et arrière de la convolution. Ainsi, le suréchantillonnage est effectué dans le réseau pour un apprentissage de bout en bout par rétropropagation à partir de la perte par pixel.

Notez que le fi ltre de déconvolution dans une telle couche n'a pas besoin d'être fi xé (par exemple, à un suréchantillonnage bilinéaire), mais peut être appris. Une pile de couches de déconvolution et de fonctions d'activation peut même apprendre un suréchantillonnage non linéaire.

Dans nos expériences, nous avons découvert que le suréchantillonnage en réseau est rapide et efficace pour l'apprentissage de la prédiction dense. Notre meilleure architecture de segmentation utilise ces couches pour apprendre à suréchantillonner pour une prédiction affinée dans la section4.2.

3.4. La formation patchwise est un échantillonnage de perte

Dans l'optimisation stochastique, le calcul du gradient est piloté par la distribution d'apprentissage. L'apprentissage par patch et l'entraînement entièrement convolutif peuvent être réalisés pour produire n'importe quelle distribution, bien que leur efficacité de calcul relative dépende du chevauchement et de la taille des minibatchs. L'apprentissage entièrement convolutif de l'image entière est identique à l'entraînement patchwise où chaque lot est constitué de tous les champs réceptifs des unités en dessous de la perte pour une image (ou une collection d'images). Bien que cela soit plus ef fi cace qu'un échantillonnage uniforme de patchs, cela réduit le nombre de lots possibles. Cependant, une sélection aléatoire de patchs dans une image peut être récupérée simplement. Restreindre la perte à un sous-ensemble échantillonné aléatoirement de ses termes spatiaux (ou appliquer de manière équivalente un masque DropConnect [39] entre la sortie et la perte) exclut les patchs du calcul du gradient.

Si les patchs conservés ont encore un chevauchement significatif, un calcul entièrement convolutif accélérera toujours l'apprentissage. Si les dégradés sont accumulés sur plusieurs passages en arrière, les lots peuvent inclure des patchs de plusieurs images.[2]

L'échantillonnage dans le cadre de la formation patchwise peut corriger le déséquilibre de classe [30, 9, 3] et atténuer la corrélation spatiale des patchs denses [31, 17]. Dans un entraînement entièrement convolutif, l'équilibre de classe peut également être obtenu en pondérant la perte, et l'échantillonnage de perte peut être utilisé pour aborder la corrélation spatiale.

Nous explorons la formation avec échantillonnage dans la section 4.3, et fait pas trouver qu'il donne une convergence plus rapide ou meilleure pour une prédiction dense. La formation à l'image entière est efficace et ef fi cace.

4. Architecture de segmentation

Nous convertissons les classi fi cateurs ILSVRC en FCN et les augmentons pour une prédiction dense avec un suréchantillonnage en réseau et une perte par pixel. Nous nous entraînons à la segmentation par réglage fin. Ensuite, nous ajoutons des sauts entre les couches pour fusionner les informations d'apparence grossière, sémantique et locale. Cette architecture de saut est apprise de bout en bout pour affiner la sémantique et la précision spatiale de la sortie.

Pour cette enquête, nous formons et validons sur le challenge de segmentation PAS-CAL VOC 2011 [8]. Nous nous entraînons avec une perte logistique multinomiale par pixel et validons avec la métrique standard de l'intersection moyenne des pixels sur l'union, la moyenne étant prise sur toutes les classes, y compris l'arrière-plan. La formation ignore les pixels masqués (comme ambigus ou dif fi ciles) dans la vérité terrain.

[2] Notez que tous les patchs possibles ne sont pas inclus de cette manière, puisque les champs de réception des unités de couche fi nale se trouvent sur une grille fi xe et quadrillée. Cependant, en décalant l'image vers la droite et vers le bas d'une valeur aléatoire jusqu'à la foulée, une sélection aléatoire de tous les patchs possibles peut être récupérée.

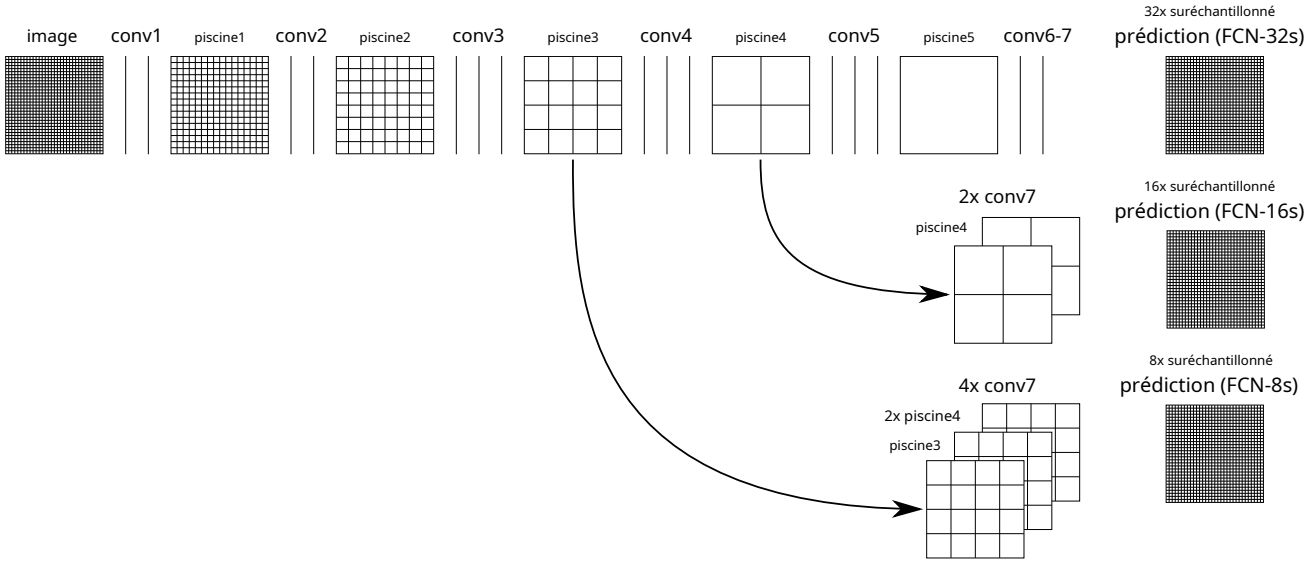

Figure 3. Nos filets DAG apprennent à combiner grossier, haut informations de couche avec des informations de couche inférieure et fines. Les couches de regroupement et de prédiction sont montré comme des grilles qui révèlent une grossièreté spatiale relative, tandis que les couches intermédiaires sont représentées sous forme de lignes verticales. Première rangée (FCN-32s): notre single-stream net, décrit dans la section 4.1, les suréchantillons ramènent 32 prédictions aux pixels en une seule étape. Deuxième rangée (FCN-16s): Combinaison des prédictions de la couche finale et de lapiscine4 La couche, à la foulée 16, permet à notre réseau de prédire les détails plus fins, tout en conservant des informations sémantiques de haut niveau. Troisième rangée (FCN-8s): prédictions supplémentaires depiscine3, à la foulée 8, apportez plus de précision.

4.1. Du classificateur au FCN dense

Nous commençons par convolutionnaliser des architectures de classi fi cation éprouvées comme dans la section 3. Nous considérons AlexNet[3] architecture [22] qui a remporté ILSVRC12, ainsi que les filets VGG [34] et le GoogLeNet[4] [35] qui a fait exceptionnellement bien dans ILSVRC14. Nous choisissons le filet à 16 couches VGG[5], ce que nous avons trouvé équivalent au réseau de 19 couches pour cette tâche. Pour GoogLeNet, nous n'utilisons que la couche de perte finale et améliorons les performances en supprimant la couche de pooling moyenne finale. Nous décapitons chaque réseau en supprimant la couche de classification finale et convertissons toutes les couches entièrement connectées en convolutions. Nous ajoutons un1 $\times$ 1 convolution avec dimension de canal 21 pour prédire les scores pour chacune des classes PAS-CAL (y compris l'arrière-plan) à chacun des emplacements de sortie grossiers, suivi d'une couche de déconvolution pour suréchantillonner bi-linéairement les sorties grossières en sorties denses en pixels comme décrit dans la section 3,3. Tableau1 compare les résultats de la validation préliminaire avec les caractéristiques de base de chaque filet. Nous rapportons les meilleurs résultats obtenus après la convergence à un taux d'apprentissage fi xe (au moins 175 époques).

Un ajustement précis de la classification à la segmentation a donné des prédictions raisonnables pour chaque réseau. Même le pire modèle réalisé ~ 75% des performances de pointe. Le réseau VGG équipé de la segmentation (FCN-VGG16) déjà

Tableau 1. Nous adaptons et étendons trois convnets de classi fi cation. Nous comparons les performances par intersection moyenne sur union sur l'ensemble de validation de PASCAL VOC 2011 et par temps d'inférence (en moyenne sur 20 essais pour un500 $\times$ 500 entrée sur une NVIDIA Tesla K40c). Nous détaillons l'architecture des réseaux adaptés en ce qui concerne la prédiction dense: nombre de couches de paramètres, taille de champ réceptif des unités de sortie, et pas le plus grossier dans le filet. (Ces chiffres donnent les meilleures performances obtenues à un taux d'apprentissage fixe, pas les meilleures performances possibles.)

	FCN-AlexNet	FCN-VGG16	FCN-GoogLeNet[4]
signifie UI	39,8	56,0	42,5
temps avant	50 ms	210 ms	59 ms
conv. couches	8	16	22
paramètres	57 M	134 M	6M
taille rf	355	404	907
foulée max	32	32	32

semble être à la pointe de la technologie avec 56,0 UI en moyenne sur val, contre 52,6 sur le test [17]. La formation sur des données supplémentaires élève FCN-VGG16 à 59,4 UI en moyenne et FCN-AlexNet à 48,0 UI en moyenne sur un sous-ensemble de val[7]. Malgré une précision de classification similaire, notre implémentation de GoogLeNet ne correspondait pas au résultat de la segmentation VGG16.

4.2. Combiner quoi et où

Nous dé fi nissons un nouveau réseau entièrement convolutif (FCN) pour la segmentation qui combine les couches de la hiérarchie d'entités et affine la précision spatiale de la sortie. Voir la figure3.

Alors que les classi fi cateurs entièrement convolutifs peuvent être

[3] Utilisation de la mise à disposition du public CaffeNet modèle de référence.

[4] Puisqu'il n'y a pas de version publique de GoogLeNet, nous utilisons notre propre réimplémentation. Notre version est entraînée avec une augmentation des données moins étendue et obtient une précision ILSVRC de 68,5% dans le top 1 et de 88,4% dans le top 5.

[5] Utilisation de la version accessible au public du zoo modèle Caffe.

réglé sur la segmentation comme indiqué dans 4.1, et même obtenir un score élevé sur la métrique standard, leur sortie est insatisfaisante (voir la figure 4). Le pas de 32 pixels au niveau de la couche de prédiction fi nale limite l'échelle des détails dans la sortie suréchantillonnée.

Nous résolvons ce problème en ajoutant des sauts [1] qui combinent la couche de prédiction finale avec les couches inférieures avec des foulées plus fines. Cela transforme une topologie de ligne en un DAG, avec des arêtes qui sautent des couches inférieures aux couches supérieures (Figure3). Comme ils voient moins de pixels, les prédictions à plus petite échelle devraient avoir besoin de moins de couches, il est donc logique de les faire à partir de sorties nettes moins profondes. La combinaison de couches fines et de couches grossières permet au modèle de faire des prédictions locales qui respectent la structure globale. Par analogie avec le jet de Koenderick et van Doorn [21], nous appelons notre hiérarchie d'entités non linéaires *jet profond.*

Nous divisons d'abord la foulée de sortie en deux en prédisant à partir d'une couche de foulée de 16 pixels. Nous ajoutons un1 $\times$1 couche de convolution au-dessus de piscine4 pour produire des prédictions de classe supplémentaires. Nous fusionnons cette sortie avec les prédictions calculées en plus deconv7 (convolutif fc7) à la foulée 32 en ajoutant un 2$\times$ couche de suréchantillonnage et sommation[6] les deux prédictions (voir la figure 3). Nous initialisons le2$\times$ suréchantillonnage en interpolation bi-linéaire, mais permet l'apprentissage des paramètres comme décrit dans la section 3,3. Enfin, les prédictions de la foulée 16 sont rééchantillonnées sur l'image. Nous appelons ce net FCN-16s. Les FCN-16 sont appris de bout en bout, initialisés avec les paramètres du dernier réseau plus grossier, que nous appelons maintenant FCN-32. Les nouveaux paramètres agissant surpiscine4 sont initialisés à zéro pour que le réseau démarre avec des prédictions non modifiées. Le taux d'apprentissage est diminué d'un facteur 100.

L'apprentissage de ce skip net améliore les performances sur la validation définie par 3,0 UI moyenne à 62,4. Chiffre4 montre une amélioration de la structure fine de la sortie. Nous avons comparé cette fusion avec l'apprentissage uniquement de lapiscine4 couche, ce qui a entraîné de mauvaises performances, et simplement une diminution du taux d'apprentissage sans ajouter le saut, ce qui a abouti à une amélioration insignifiante des performances sans améliorer la qualité de la sortie.

Nous continuons de cette façon en fusionnant les prédictions de piscine3 avec un 2$\times$ suréchantillonnage des prédictions fusionnées à partir de piscine4 et conv7, construire les FCN-8 nets. Nous obtenons une amélioration supplémentaire mineure à 62,7 UI moyens, et trouvons une légère amélioration de la finesse et du détail de notre sortie. À ce stade, nos améliorations de fusion ont rencontré des rendements diminishing, à la fois en ce qui concerne la métrique IU qui met l'accent sur l'exactitude à grande échelle, et aussi en termes de l'amélioration visible par exemple dans la figure4, nous ne continuons donc pas à fusionner des couches encore plus basses.

Raffinement par d'autres moyens Diminuer la foulée de la mise en commun des couches est le moyen le plus simple d'obtenir des prédictions plus fines. Cependant, cela pose problème pour notre réseau basé sur VGG16. Réglage dupiscine5 pas à 1 nécessite notre convolutionnalisé fc6 avoir la taille du noyau 14 $\times$ 14 à

[6]La fusion maximale a rendu l'apprentissage difficile en raison de la commutation de gradient.

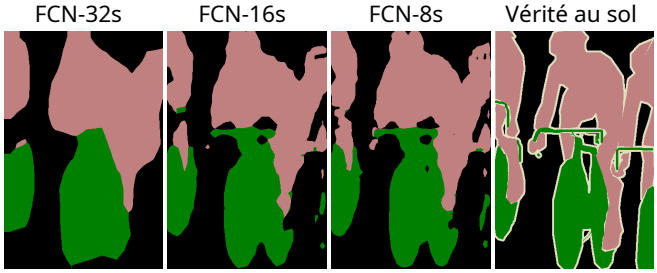

Figure 4. Le raffinement des réseaux entièrement convolutifs en fusionnant les informations des couches avec des pas différents améliore les détails de la segmentation. Les trois premières images montrent la sortie de nos filets de pas de 32, 16 et 8 pixels (voir Figure3).

Tableau 2. Comparaison des sauts de FCN sur un sous-ensemble[7] de PASCAL VOC 2011 segval. L'apprentissage se fait de bout en bout, sauf pour le FCN-32s fi xé, où seule la dernière couche est finement réglée. Notez que FCN-32s est FCNVGG16, renommé pour mettre en évidence la foulée.

	pixel fw	moyenne acc.	moyenne UI	UI
FCN-32s- fi xé	83,0	59,7	45,4	72,0
FCN-32s	89,1	73,3	59,4	81,4
FCN-16s	90,0	75,7	62,4	83,0
FCN-8s	90,3	75,9	62,7	83,2

maintenir sa taille de champ réceptif. En plus de leur coût de calcul, nous avons eu des difficultés à apprendre de si gros fi ltres. Nous avons tenté de réorganiser les couches ci-dessuspiscine5 avec des fi ltres plus petits, mais n'a pas atteint des performances comparables; une explication possible est que l'initialisation ILSVRC des couches supérieures est importante.

Une autre façon d'obtenir des prédictions plus fines consiste à utiliser l'astuce shift-and-stitch décrite dans la section 3.2. Dans des expériences limitées, nous avons trouvé que le rapport coût / amélioration de cette méthode était pire que la fusion de couches.

4.3. Cadre expérimental

Optimisation Nous nous entraînons avec SGD avec élan. Nous utilisons une taille de mini-lot de 20 images et des taux d'apprentissage fi xes de dix^{-3}, dix^{-4}, et 5^{-5} pour FCN-AlexNet, FCN-VGG16 et FCN-GoogLeNet, respectivement, choisis par recherche de ligne. Nous utilisons l'élan0,9, décroissance de poids 5^{-4} ou alors 2^{-4}, et le taux d'apprentissage doublé pour les biais, bien que nous ayons constaté que la formation était sensible au seul taux d'apprentissage. Nous initialisons à zéro la couche de score de classe, car l'initialisation aléatoire n'a donné ni de meilleures performances ni une convergence plus rapide. L'abandon était inclus là où il était utilisé dans les filets de classification d'origine.

Réglage fin Nous affinons toutes les couches par rétropropagation à travers tout le réseau. Affiner le Le classificateur de sortie à lui seul ne donne que 70% des performances de réglage complètes par rapport au tableau 2. La formation à partir de zéro n'est pas faisable compte tenu du temps nécessaire pour apprendre les réseaux de classification de base. (Notez que le réseau VGG est formé par étapes, tandis que nous initialisons à partir de la couche 16

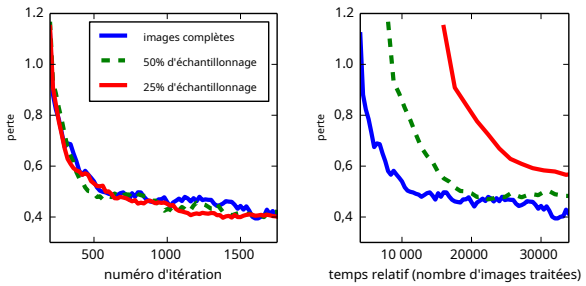

Figure 5. L'entraînement sur des images entières est tout aussi efficace que l'échantillonnage de patchs, mais se traduit par une convergence plus rapide (temps de mur) en utilisant plus efficacement les données. La gauche montre l'effet de l'échantillonnage sur le taux de convergence pour une taille de lot attendue fi xe, tandis que la droite le représente en fonction du temps de paroi relatif.

Le réglage fin prend trois jours sur un seul GPU pour la version grossière des FCN-32, et environ un jour chacun pour passer aux versions FCN-16 et FCN-8.

Plus de données d'entraînement Le coffret de formation à la segmentation PASCAL VOC 2011 étiquette 1112 images. Hariharan*et coll. [*16] a collecté des étiquettes pour un plus grand ensemble de 8498 images de formation PASCAL, qui ont été utilisées pour former l'ancien système de pointe, SDS [17]. Ces données d'entraînement améliorent le score de validation FCN-VGG16[7] de 3,4 points à 59,4 UI moyenne.

Échantillonnage de patch Comme expliqué dans la section 3.4, notre plein La formation d'image regroupe efficacement chaque image dans une grille régulière de grands patchs qui se chevauchent. En revanche, des travaux antérieurs échantillonnent aléatoirement des patchs sur un ensemble de données complet [30, 3, 9, 31, 11], ce qui pourrait entraîner des lots de variance plus élevés susceptibles d'accélérer la convergence [24]. Nous étudions ce compromis en échantillonnant spatialement la perte de la manière décrite précédemment, en faisant un choix indépendant d'ignorer chaque cellule de la couche finale avec une certaine probabilité1-*p*. Pour éviter de changer la taille effective du lot, nous augmentons simultanément le nombre d'images par lot d'un facteur 1/*p*. Notez qu'en raison de l'efficacité de la convolution, cette forme d'échantillonnage de rejet est toujours plus rapide que l'apprentissage par patch pour des valeurs suffisamment grandes de *p* (par exemple, au moins pour *p* > 0,2 selon les nombres de la section 3,1). Chiffre5 montre l'effet de cette forme d'échantillonnage sur la convergence. Nous constatons que l'échantillonnage n'a pas d'effet signi fi catif sur le taux de convergence par rapport à la formation d'image entière, mais prend beaucoup plus de temps en raison du plus grand nombre d'images à prendre en compte par lot. Nous choisissons donc une formation d'image entière non échantillonnée dans nos autres expériences.

Équilibrage des classes Un entraînement entièrement convolutif peut équilibrer les classes en pondérant ou en échantillonnant la perte. Bien que nos étiquettes soient légèrement déséquilibrées (environ3/4 sont en arrière-plan), nous trouvons que l'équilibrage des classes n'est pas nécessaire.

Prédiction dense Les scores sont suréchantillonnés à l'entrée dimensions par couches de déconvolution dans le réseau. Les fi ltres déconvolutionnels de la couche finale sont fi xés à une interpolation bilinéaire, tandis que les couches intermédiaires de suréchantillonnage sont initialisées à un suréchantillonnage bilinéaire, puis apprises.

Augmentation Nous avons essayé d'augmenter les données d'apprentissage en reflétant et en «jitter» aléatoirement les images en les traduisant jusqu'à 32 pixels (l'échelle de prédiction la plus grossière) dans chaque direction. Cela n'a donné aucune amélioration notable.

Mise en œuvre Tous les modèles sont formés et testés avec Caffe [20] sur un seul NVIDIA Tesla K40c. Nos modèles et notre code sont accessibles au public sur http://fcn.berkeleyvision.org.

5. Résultats

Nous testons notre FCN sur la segmentation sémantique et l'analyse de scène, en explorant PASCAL VOC, NYUDv2 et SIFT Flow. Bien que ces tâches aient historiquement distingué les objets et les régions, nous traitons les deux uniformément comme une prédiction de pixels. Nous évaluons notre architecture de saut FCN sur chacun de ces ensembles de données, puis l'étendons à l'entrée multimodale pour NYUDv2 et à la prédiction multi-tâches pour les étiquettes sémantiques et géométriques de SIFT Flow.

Métrique Nous rapportons quatre métriques issues d'évaluations communes de segmentation sémantique et d'analyse de scène qui varient tions sur $\sum$précision des pixels et intersection des régions sur l'union (UI). Laisser n_{ij} être le nombre de pixels de la classe *je* devrait appartenir à la classe *j,* où il y a *n* différentes classes, et laisser $t_i = \sum_j n_{ij}$ être le tot$\sum$ nombre de pixels de classe *je.* Nous calculer:

- précision des pixels: $\sum_{je} n_{ii} / (\sum_{je} t_{je})$
- précision moyenne cy: $(1/n_{cl}) \sum_{je} (n_{ii} / t_{je})$
- signifie UI: $(1/n_{cl}) \sum_{je} n_{ii} / (t_{je} + \sum_j n_{ji} - n_{ii})$
- la fréquence$\sum$poids ed UI$\sum$ $(\sum_k t_k)^{-1} \sum_{je} t_{je} n_{ii} / (t_{je} + \sum_j n_{ji} - n_{ii})$

COV PASCAL Tableau 3 donne le performance de notre FCN-8 sur les ensembles de test de PASCAL VOC 2011 et 2012, et le compare à l'état de l'art antérieur, SDS [17] et le célèbre R-CNN [12]. Nous obtenons les meilleurs résultats sur l'IU moyenne[8] par une marge relative de 20%. Le temps d'inférence est réduit114× (convnet uniquement, ignorant les propositions et les raffinements) ou 286× (globalement).

Tableau 3. Notre réseau entièrement convolutif donne une amélioration relative de 20% par rapport à l'état de l'art sur le PASCALVOC 2011 et 2012 ensembles de test et réduit le temps d'inférence.

	signifie UI Test VOC2011	signifie UI Test VOC2012	inférence temps
R-CNN [12]	47,9	-	-
FDS [17]	52,6	51,6	~50 s
FCN-8s	62,7	62,2	~175 ms

[7] Il y a des images d'entraînement de [16] inclus dans le jeu de valeurs PASCAL VOC 2011, nous validons donc sur le jeu de 736 images sans intersection. Une version antérieure de cet article a été évaluée par erreur sur l'ensemble des valeurs.

[8] C'est la seule métrique fournie par le serveur de test.

Tableau 4. Résultats sur NYUDv2. *RGBD* est la fusion précoce des canaux RVB et de profondeur à l'entrée. *HHA* est l'incorporation en profondeur de [15] en tant que disparité horizontale, hauteur au-dessus du sol et angle de la normale de la surface locale avec la direction de gravité déduite. *RVB-HHA* est le modèle de fusion tardive formé conjointement qui additionne les prédictions RVB et HHA.

	pixel acc.	moyenne acc.	moyenne UI	fw UI
Gupta *et coll. [*15	60,3	-	28,6	47,0
] FCN-32s RVB	60,0	42,2	29,2	43,9
RGBD FCN-32s	61,5	42,4	30,5	45,5
FCN-32s HHA	57,1	35,2	24,2	40,4
FCN-32s RGB-HHA	64,3	44,9	32,8	48,0
FCN-16s RGB-HHA	65,4	46,1	34,0	49,5

NYUDv2 [33] est un ensemble de données RVB-D collecté à l'aide de Microsoft Kinect. Il a 1449 images RVB-D, avec des étiquettes en pixels qui ont été fusionnées en une tâche de segmentation sémantique de 40 classes par Gupta*et coll. [*14]. Nous rapportons les résultats sur la division standard de 795 images d'entraînement et 654 images de test. (Remarque: toute la sélection de modèle est effectuée sur la valeur PAS-CAL 2011.) Tableau4 donne les performances de notre modèle en plusieurs variantes. Nous formons d'abord notre modèle grossier non modifié (FCN-32s) sur des images RVB. Pour ajouter des informations de profondeur, nous nous entraînons sur un modèle mis à niveau pour prendre une entrée RVB-D à quatre canaux (fusion précoce). Cela n'apporte que peu d'avantages, peut-être en raison de la difficulté de propager des gradients significatifs tout au long du modèle. Suite au succès de Gupta*et coll. [*15], nous essayons le codage HHA tridimensionnel de la profondeur, entraînant les filets uniquement sur ces informations, ainsi qu'une «fusion tardive» de RVB et HHA où les prédictions des deux réseaux sont additionnées à la couche finale, et le résultat Le réseau à deux flux est appris de bout en bout. Enfin, nous mettons à niveau ce filet de fusion tardif vers une version à 16 foulées.

Flux SIFT est un ensemble de données de 2 688 images avec des étiquettes de pixels pour 33 catégories sémantiques («pont», «montagne», «soleil»), ainsi que trois catégories géométriques («horizontale», «verticale» et «ciel»). Un FCN peut naturellement apprendre une représentation conjointe qui prédit simultanément les deux types d'étiquettes. Nous apprenons une version à deux têtes des FCN-16 avec des couches et des pertes de prédiction sémantique et géométrique. Le modèle appris fonctionne aussi bien sur les deux tâches que sur deux modèles formés indépendamment, tandis que l'apprentissage et l'inférence sont essentiellement aussi rapides que chaque modèle indépendant en lui-même. Les résultats dans le tableau5, calculé sur le standard divisé en 2488 images d'entraînement et 200 images de test,[9] montrer des performances de pointe sur les deux tâches.

[9]Trois des catégories de flux SIFT ne sont pas présentes dans l'ensemble de test. Nous avons fait des prédictions pour les 33 catégories, mais nous n'avons inclus que les catégories réellement présentes dans l'ensemble de test dans notre évaluation. (Une version antérieure de ce papier rapportait une UI moyenne plus faible, qui comprenait toutes les catégories présentes ou prévues dans l'évaluation.)

Tableau 5. Résultats sur SIFT Flow[9] avec segmentation de classe (centre) et segmentation géométrique (à droite). Tighe [36] est une méthode de transfert non paramétrique. Tighe 1 est un SVM exemplaire tandis que 2 est SVM + MRF. Farabet est un convnet multi-échelles formé sur des échantillons à classe équilibrée (1) ou des échantillons de fréquence naturelle (2). Pinheiro est un convnet récurrent multi-échelles, de-
c'est noté RCNN3 (◦3). La métrique car la géométrie est la précision des pixels.

	pixel acc.	moyenne acc.	moyenne UI	fw UI	geom. acc.
Liu *et coll. [*25]	76,7	-	-	-	-
Tighe *et coll. [*36]	-	-	-	-	90,8
Tighe *et coll. [*37] 1	75,6	41,1	-	-	-
Tighe *et coll. [*37] 2	78,6	39,2	-	-	-
Farabet *et coll. [*9] 1	72,3	50,8	-	-	-
Farabet *et coll. [*9] 2	78,5	29,6	-	-	-
Pinheiro *et coll. [*31]	77,7	29,8	-	-	-
FCN-16s	85,2	51,7	39,5	76,1	94,3

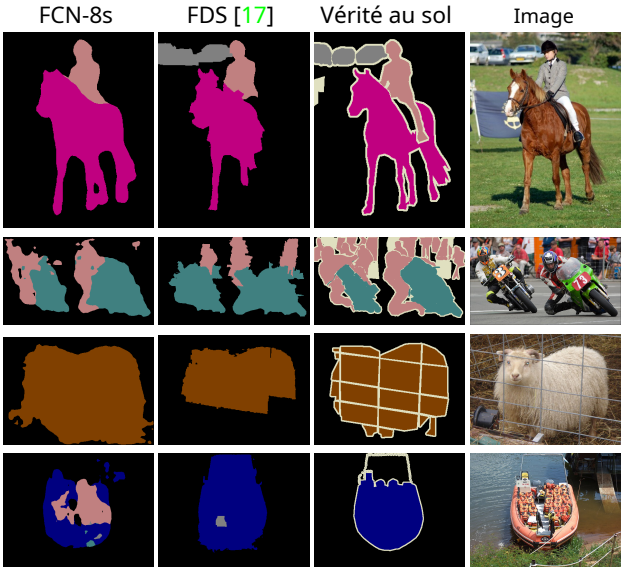

Figure 6. Entièrement convolutif les réseaux de segmentation produisent des états performance de pointe sur PASCAL. La colonne de gauche montre la sortie de notre réseau le plus performant, les FCN-8. La seconde montre les segmentations produites par l'ancien système de pointe de Hariharan*et coll. [*17]. Remarquez les fines structures récupérées (première ligne), la capacité de séparer les objets en interaction étroite (deuxième ligne) et la robustesse aux occluseurs (troisième ligne). La quatrième ligne montre un cas d'échec: le filet voit les gilets de sauvetage dans un bateau comme des personnes.

6. Conclusion

Les réseaux entièrement convolutifs sont une classe riche de modèles, dont les convnets de classi fi cation modernes sont un cas particulier. Conscient de cela, étendre ces réseaux de classification à la segmentation et améliorer l'architecture avec des combinaisons de couches multi-résolution améliore considérablement l'état de l'art, tout en simplifiant et en accélérant simultanément l'apprentissage et l'inférence.

Remerciements Ce travail a été soutenu en partie

par les programmes MSEE et SMISC de la DARPA, la NSF récompense l'IIS-1427425, IIS-1212798, IIS-1116411 et le NSF GRFP, Toyota et le Berkeley Vision and Learning Center. Nous remercions vivement NVIDIA pour son don de GPU. Nous remercions Bharath Hariharan et Saurabh Gupta pour leurs conseils et leurs outils de jeu de données. Nous remercions Sergio Guadarrama d'avoir reproduit GoogLeNet dans Caffe. Nous remercions Jitendra Malik pour ses précieux commentaires. Merci à Wei Liu pour avoir signalé un problème avec notre calcul SIFT Flow Mean IU et une erreur dans notre formule IU moyenne pondérée en fréquence.

A. Limites supérieures de l'IU

Dans cet article, nous avons obtenu de bonnes performances sur la métrique de segmentation moyenne des UI, même avec une prédiction sémantique grossière. Pour mieux comprendre cette métrique et les limites de cette approche par rapport à elle, nous calculons des limites supérieures approximatives sur les performances avec prédiction à différentes échelles. Nous faisons cela en sous-échantillonnant les images de vérité terrain, puis en les suréchantillonnant à nouveau pour simuler les meilleurs résultats pouvant être obtenus avec un facteur de sous-échantillonnage particulier. Le tableau suivant donne l'IU moyenne sur un sous-ensemble de la valeur PASCAL 2011 pour divers facteurs de sous-échantillonnage.

facteur	signifie UI
128	50,9
64	73,3
32	86,1
16	92,8
8	96,4
4	98,5

Prédiction parfaite au pixel près n'est clairement pas nécessaire de atteindre une UI moyenne bien au-dessus de l'état de la technique et, inversement, une UI moyenne n'est pas une bonne mesure de la précision à échelle fine.

B. Plus de résultats

Nous évaluons en outre notre FCN pour la segmentation sémantique.

PASCAL-Contexte [29] fournit des annotations de scènes entières de PASCAL VOC 2010. Bien qu'il existe plus de 400 classes distinctes, nous suivons la tâche de 59 classes dé fi nie par [29] qui sélectionne les cours les plus fréquents. Nous formons et évaluons respectivement sur les ensembles de formation et de valorisation. Dans le tableau6, nous comparons à la variation objet joint + bourrage du masquage de fonction convolutive [4] qui est l'état de l'art antérieur sur cette tâche. Les scores de FCN-8s sont de 37,8 UI en moyenne pour une amélioration relative de 20%.

Tableau 6. Résultats sur PASCAL-Contexte. *CFM* est le meilleur résultat de [4] par masquage de caractéristiques convolutives et poursuite de segment avec le VGG net. *O₂P* est la méthode de mise en commun du second ordre [2] comme indiqué dans le *errata* de [29]. La tâche de 59 classes sélectionne les 59 plus fréquentes classes d'évaluation.

59 cours	pixel acc.	moyenne acc.	moyenne UI	fw UI
O₂P	-	-	18,1	-
CFM	-	-	31,5	-
FCN-32s	65,4	47,2	35,1	50,3
FCN-16s	66,8	49,6	37,6	52,3
FCN-8s	67,0	50,7	37,8	52,5

Journal des modifications

La version arXiv de ce document est mise à jour avec des corrections et des informations supplémentaires pertinentes. Ce qui suit donne un bref historique des changements.

v2 Ajouter une annexe UNE donner des limites supérieures sur une UI moyenne et appendice B avec les résultats PASCAL-Context. Corriger les nombres de validation PAS-CAL (auparavant, certaines images val étaient incluses dans le train), les UI moyennes de flux SIFT (qui utilisaient une métrique insuffisamment stricte) et une erreur dans la formule de la moyenne pondérée en fréquence des UI. Ajoutez un lien vers les modèles et mettez à jour les numéros de synchronisation pour refléter une mise en œuvre améliorée (qui est accessible au public).

Les références

[1] CM Bishop. *Reconnaissance de formes et apprentissage automatique,* page 229. Springer-Verlag New York, 2006. 6

[2] J. Carreira, R. Caseiro, J. Batista et C. Sminchisescu. Segmentation sémantique avec mise en commun de second ordre. Dans*ECCV,* 2012. 9

[3] DC Ciresan, A. Giusti, LM Gambardella et J. Schmidhuber. Les réseaux de neurones profonds segmentent les membranes neuronales dans les images de microscopie électronique. Dans*NIPS,* pages 2852-2860, 2012. 1, 2, 4, 7

[4] J. Dai, K. He et J. Sun. Masquage des fonctions convolutives pour la segmentation des objets et des objets joints.*préimpression arXiv arXiv: 1412.1283,* 2014. 9

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng et T. Darrell. DeCAF: une fonction d'activation convolutionnelle profonde pour la reconnaissance visuelle générique. Dans*ICML,* 2014. 1, 2

[6] D. Eigen, D. Krishnan et R. Fergus. Restauration d'une image prise à travers une fenêtre recouverte de saleté ou de pluie. Dans *Computer Vision (ICCV), 2013 IEEE International Conference on,* pages 633–640. IEEE, 2013.2

[7] D. Eigen, C. Puhrsch et R. Fergus. Prédiction de la carte de profondeur à partir d'une seule image à l'aide d'un réseau profond à plusieurs échelles. *préimpression arXiv arXiv: 1406.2283,* 2014. 2

[8] M. Everingham, L. Van Gool, CKI Williams, J. Winn et A. Zisserman. Résultats du PASCAL Visual Object Classes Challenge 2011 (VOC2011). http://www.pascalnetwork.org/challenges/VOC/voc2011/workshop/index.html. 4

[9] C. Farabet, C. Couprie, L. Najman et Y. LeCun. Apprentissage des fonctionnalités hiérarchiques pour l'étiquetage des scènes.*Analyse de modèles et intelligence artificielle, transactions IEEE sur,* 2013. 1, 2, 4, 7, 8

[10] P. Fischer, A. Dosovitskiy et T. Brox. Correspondance des descripteurs avec les réseaux de neurones convolutifs: une comparaison avec SIFT. *CoRR,* abs / 1405.5769, 2014. 1

[11] Y. Ganin et V. Lempitsky. N4-champs: champs les plus proches du réseau neuronal pour les transformations d'image. Dans*ACCV,* 2014. 1, 2, 7

[12] R. Girshick, J. Donahue, T. Darrell et J. Malik. Hiérarchies de fonctionnalités riches pour une détection précise des objets et une segmentation sémantique. Dans*Vision par ordinateur et reconnaissance de formes,* 2014. 1, 2, 7

[13] A. Giusti, DC Cireşan, J. Masci, LM Gambardella et J. Schmidhuber. Balayage d'image rapide avec des réseaux de neurones convolutifs à pool maximal profond. Dans*ICIP,* 2013. 3, 4

[14] S. Gupta, P. Arbelaez et J. Malik. Organisation perceptuelle et reconnaissance des scènes d'intérieur à partir d'images RVB-D. Dans *CVPR,* 2013. 8

[15] S. Gupta, R. Girshick, P. Arbelaez et J. Malik. Apprentissage de fonctionnalités riches à partir d'images RVB-D pour la détection et la segmentation d'objets. Dans*ECCV.* Springer, 2014. 1, 2, 8

[16] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji et J. Malik. Contours sémantiques des détecteurs inverses. Dans*Conférence internationale sur la vision par ordinateur (ICCV),* 2011. 7

[17] B. Hariharan, P. Arbeláez, R. Girshick et J. Malik. Détection et segmentation simultanées. Dans*Conférence européenne sur la vision par ordinateur (ECCV),* 2014. 1, 2, 4, 5, 7, 8

[18] B. Hariharan, P. Arbeláez, R. Girshick et J. Malik. Hypercolonnes pour la segmentation d'objets et la localisation fine. Dans*Vision par ordinateur et reconnaissance de formes,* 2015. 2

[19] K. He, X. Zhang, S. Ren et J. Sun. Regroupement de pyramides spatiales dans des réseaux convolutifs profonds pour la reconnaissance visuelle. Dans *ECCV,* 2014. 1, 2

[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama et T. Darrell. Caffe: architecture convolutionnelle pour une intégration rapide des fonctionnalités.*préimpression arXiv arXiv: 1408.5093,* 2014. 7

[21] JJ Koenderink et AJ van Doorn. Représentation de la géométrie locale dans le système visuel.*Cybernétique biologique,* 55 (6): 367–375, 1987. 6

[22] A. Krizhevsky, I. Sutskever et GE Hinton. Classi fi cation Imagenet avec des réseaux de neurones convolutifs profonds. Dans *NIPS,* 2012. 1, 2, 3, 5

[23] Y. LeCun, B. Boser, J. Denker, D. Henderson, RE Howard, W. Hubbard et LD Jackel. Rétropropagation appliquée à la reconnaissance manuscrite du code postal. Dans*Calcul neuronal,* 1989. 2, 3

[24] YA LeCun, L. Bottou, GB Orr et K.-R. Müller. Backprop efficace. Dans*Réseaux de neurones: astuces du métier,* pages 9–48. Springer, 1998.7

[25] C. Liu, J. Yuen et A. Torralba. Sift fl ow: correspondance dense entre les scènes et ses applications.*Analyse de modèles et intelligence artificielle, transactions IEEE sur,* 33 (5): 978–994, 2011. 8

[26] J. Long, N. Zhang et T. Darrell. Les convnets apprennent-ils la correspondance? Dans*NIPS,* 2014. 1

[27] S. Mallat. *Une visite en ondelettes du traitement du signal.* Presse académique, 2e édition, 1999. 4

[28] O. Matan, CJ Burges, Y. LeCun et JS Denker. Reconnaissance multidigit à l'aide d'un réseau de neurones à déplacement spatial. Dans*NIPS,* pages 488–495. Citeseer, 1991.2

[29] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun et A. Yuille. Le rôle du contexte pour la détection d'objets et la segmentation sémantique à l'état sauvage. Dans*Computer Vision and Pattern Recognition (CVPR), Conférence IEEE 2014 sur,* pages 891–898. IEEE, 2014.9

[30] F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou et PE Barbano. Vers le phénotypage automatique des embryons en développement à partir de vidéos.*Traitement d'image, transactions IEEE activées,* 14 (9): 1360-1371, 2005. 1, 2, 4, 7

[31] PH Pinheiro et R. Collobert. Réseaux de neurones convolutifs récurrents pour l'étiquetage de scènes. Dans*ICML,* 2014. 1, 2, 4, 7, 8

[32] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus et Y. LeCun. Overfeat: reconnaissance, localisation et détection intégrées à l'aide de réseaux convolutifs. Dans*ICLR,* 2014. 1, 2, 4

[33] N. Silberman, D. Hoiem, P. Kohli et R. Fergus. Segmentation intérieure et inférence de support à partir d'images rgbd. Dans *ECCV,* 2012. 8

[34] K. Simonyan et A. Zisserman. Très profonde convolu- réseaux internationaux pour la reconnaissance d'images à grande échelle. *CoRR,* abs / 1409.1556, 2014. 1, 2, 3, 5

[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke et A. Rabinovich. Aller plus loin avec les circonvolutions.*CoRR,* abs / 1409,4842, 2014. 1, 2, 3, 5

[36] J. Tighe et S. Lazebnik. Superparsing: analyse d'image non paramétrique évolutive avec des superpixels. Dans*ECCV,* pages 352–365. Springer, 2010. 8

[37] J. Tighe et S. Lazebnik. Recherche de choses: analyse d'images avec des régions et des détecteurs par exemplaire. Dans*CVPR,* 2013. 8

[38] J. Tompson, A. Jain, Y. LeCun et C. Bregler. Formation conjointe d'un réseau convolutif et d'un modèle graphique pour l'estimation de la pose humaine.*CoRR,* abs / 1406.2984, 2014. 2

[39] L. Wan, M. Zeiler, S. Zhang, YL Cun et R. Fergus. Régularisation des réseaux de neurones à l'aide de dropconnect. Dans*Actes de la 30e Conférence internationale sur l'apprentissage automatique (ICML-13),* pages 1058 à 1066, 2013. 4

[40] R. Wolf et JC Platt. Emplacement du bloc d'adresse postale à l'aide d'un réseau de localisateurs convolutifs.*Progrès dans les systèmes de traitement de l'information neuronale,* pages 745–745, 1994. 2

[41] MD Zeiler et R. Fergus. Visualiser et comprendre les réseaux convolutifs. Dans*Vision par ordinateur - ECCV 2014,* pages 818–833. Springer, 2014.2

[42] N. Zhang, J. Donahue, R. Girshick et T. Darrell. R-cnns partiels pour la détection de catégorie à granularité fine. Dans*Vision par ordinateur - ECCV 2014,* pages 834–849. Springer, 2014. 1