BRÈVE COMMUNICATION

**https://doi.org/10.1038/s41592-018-0261-2**

[**Corrigé : Correction de l’auteur**](https://doi.org/10.1038/s41592-019-0356-4)

**U-Net : apprentissage profond pour le comptage, la détection et la morphométrie des cellules**

**Thorsten Falk1,2,3,20, Dominic Mai1,2,4,16,20, Robert Bensch1,2,17,20, Özgün Çiçek1, Ahmed Abdulkadir1,5,**

**Yassine Marrakchi1,2,3, Anton Böhm1, Jan Deubner6,7, Zoe Jäckel6,7, Katharina Seiwald6,**

**Alexander Dovzhenko8,18, Olaf Tietz8,18, Cristina Dal Bosco8, Sean Walsh8,18, Deniz Saltukoglu2,9,10,11,**

**Tuan Leng Tay7,12,13, Marco Prinz2,3,12, Klaus Palme2,8, Matias Simons2,9,10,14, Ilka Diester6,7,15,**

**Thomas Brox1,2,3,7 et Olaf Ronneberger1,2,19\***

**U-Net est une solution d’apprentissage approfondie générique pour les tâches de quantification fréquentes telles que la détection de cellules et les mesures de forme dans les données d’images biomédicales. Nous présentons un plugin ImageJ qui permet aux experts en apprentissage non-machine d’analyser leurs données avec U-Net sur un ordinateur local ou un serveur distant / service cloud. Le plugin est livré avec des modèles pré-remplis pour la segmentation à une seule cellule et permet à U-Net d’être adapté à de nouvelles tâches sur la base de quelques échantillons annotés.**

Les progrès des techniques de microscopie et de préparation des échantillons ont laissé aux chercheurs de grandes quantités de données d’image. Ces données promettraient des informations supplémentaires, une analyse plus précise et des statistiques plus rigoureuses, si ce n’est pour les obstacles de la quantification. Les images doivent d’abord être converties en nombres avant que l’analyse statistique puisse être appliquée. Souvent, cette conversion nécessite de compter des milliers de cellules avec certains marqueurs ou de dessiner les contours des cellules pour quantifier leur forme ou la force d’un reporter. Ce travail est fastidieux et par conséquent souvent évité. Dans les études neurostifiques utilisant des outils optogénétiques, par exemple, la quantification du nombre de neurones exprimant l’opsine ou la localisation des opsines nouvellement développées dans les cellules est souvent demandée. Toutefois, en raison des efforts requis, la plupart des études sont publiées sans cette information.

Une telle quantification n’est-elle pas un travail que les ordinateurs peuvent faire? En effet. Depuis des décennies, les informaticiens mettent au point des logiciels spécialisés qui peuvent alléger le fardeau de la quantification des chercheurs en sciences de la vie. Cependant, chaque laboratoire produit des données différentes et se concentre sur différents aspects des données pour la question de recherche à l’étude. Ainsi, un nouveau logiciel doit être construit pour chaque cas. L’apprentissage profond pourrait changer cette situation. Parce que l’apprentissage profond apprend les caractéristiques pertinentes pour la tâche à partir de données plutôt que d’être codé en dur, les ingénieurs logiciels n’ont pas besoin de configurer un logiciel spécialisé pour une certaine tâche de quantification. Au lieu de cela, un progiciel générique peut apprendre à s’adapter à la tâche de façon autonome à partir de données appropriées, des données que les chercheurs en sciences de la vie peuvent eux-mêmes fournir.

Les approches fondées sur l’apprentissage ont suscité l’intérêt de la communauté biomédicale il y a des années. Des solutions couramment utilisées, telles que ilastik1 (http://ilastik.org/[) ou le kit d’outils de segmentation WEKA](http://ilastik.org/) formatable2 (https://imagej.net/Trainable\_Weka\_Segmentation[) permettent de former les pipelines de segmentation en utilisant des fonctions d’image génériques personnalisées. Plus récemment, l’attention s’est tournée vers l’apprentissage en profondeur, qui extrait automatiquement les caractéristiques optimales pour la tâche d’analyse d’images, éliminant ainsi le besoin de conception de fonctionnalités par des experts en sciences de l’information3-5. Cependant, l’utilisation généralisée de la quantification dans les sciences de la vie a été entravée par l’absence de progiciels génériques faciles à utiliser. Bien que les progiciels Aivia (](https://imagej.net/Trainable_Weka_Segmentation)https:// www.drvtechnologies.com/aivia6/) et Cell Profiler6 [(http://cell-](https://www.drvtechnologies.com/aivia6/) [profiler.org/](https://www.drvtechnologies.com/aivia6/)) utilisent déjà des modèles d’apprentissage approfondi, ils ne permettent pas de formation sur les nouvelles données, limitant ainsi leur domaine d’application à un petit nombre d’ensembles de données. Dans le cadre de la restauration d’images, CSBDeep7 [fournit un plugin ImgLib2 (](http://cellprofiler.org/)https://imagej.net/ImgLib2) avec des modèles pour des modalités d’imagerie spécifiques et des spécimens biologiques, et permet l’intégration de nouveaux modèles de restauration formés à l’externe. CDeep3M8 est une autre approche de l’apprentissage profond des sciences de la vie, qui fournit un ensemble d’outils de ligne de commande et de tutoriels pour la formation et l’application d’une architecture résiduelle de création de réseau pour la segmentation d’images 3D. Ces progiciels répondent spécifiquement aux besoins occasionnels des chercheurs en matière d’apprentissage profond en fournissant une configuration basée sur le cloud qui ne nécessite pas d’unité locale de traitement graphique (GPU).

Le présent travail fournit un progiciel générique basé sur l’apprentissage profond pour la détection et la segmentation des cellules. Pour notre architecture de réseau d’encodeur-décodeur U-Net3, qui a déjà atteint les premiers rangs dans les benchmarks d’analyse de données biomédicales9 et qui a été la base de nombreux modèles d’apprentissage profond en analyse d’images biomédicales, nous avons développé une interface qui fonctionne comme un plu-dans le logiciel ImageJ couramment utilisé10 (Note supplémentaire 1 et Logiciel supplémentaire 1–4). Contrairement à tous les progiciels précédents en son genre, notre U-Net peut être formé et adapté pour

1Département d’informatique, Université Albert-Ludwigs, Fribourg, Allemagne. 2BIOSS Centre for Biological Signalling Studies, Fribourg, Allemagne.

3CIBSS Centre for Integrative Biological Signalling Studies, Université Albert-Ludwigs, Fribourg, Allemagne. 4Life Imaging Center, Center for Biological Systems Analysis, Université Albert-Ludwigs, Fribourg, Allemagne. 5University Hospital of Old Age Psychiatry and Psychotherapy, Université de Berne, Berne, Suisse. 6Laboratoire d’Optophysiologie, Institut de biologie III, Université Albert-Ludwigs, Fribourg, Allemagne. 7BrainLinks-BrainTools, Albert-Ludwigs-University, Fribourg, Allemagne. 8Institute of Biology II, Albert-Ludwigs-University, Fribourg, Allemagne. 9Center for Biological Systems Analysis (ZBSA), Albert-Ludwigs-University, Fribourg, Allemagne. 10Renal Division, University Medical Centre, Fribourg, Allemagne. 11Spemann Graduate School of Biology and Medicine (SGBM), Albert-Ludwigs-University, Fribourg, Allemagne. 12Institute of Neuropathology, University Medical Centre, Fribourg, Allemagne.

13Institut de biologie I, Université Albert-Ludwigs, Fribourg, Allemagne. 14Paris Descartes University-Sorbonne Paris Cité, Institut Imagine, Paris, France.

15Bernstein Center Freiburg, Albert-Ludwigs-University, Freiburg, Allemagne. 16Adresse de presse : SICK AG, Waldkirch, Allemagne. 17Adresse de presse : ANavS GmbH,

München, Allemagne. 18Adresse de presse : ScreenSYS GmbH, Freiburg, Allemagne. 19Adresse de presse : DeepMind, Londres, Royaume-Uni. 20Ces auteurs ont contribué

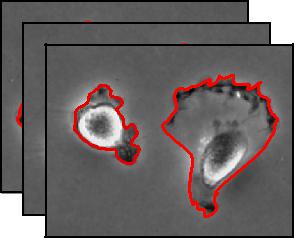
également : Thorsten Falk, Dominic Mai, Robert Bensch. \*e-mail : [ronneber@informatik.uni-freiburg.de](mailto:ronneber@informatik.uni-freiburg.de)

|  |  |
| --- | --- |
| **NATURE METHODS** | VOL 16 | JANVIER 2019 | 67–70 [| www.nature.com/naturemethods](http://www.nature.com/naturemethods) | **67** |

BRÈVE COMMUNICATION **MÉTHODES NATURE**

Formation/transfert d’apprentissage (une seule fois)

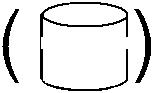
**un**



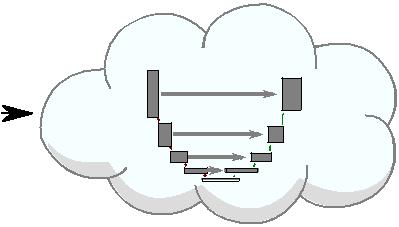
Images avec annotation de ROI

* Préformé

modèle

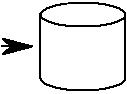


Service de serveur ou de nuage



U-Net

Adapté



modèle

**b**



Service de serveur ou de nuage



Images brutes

+

Masques de segmentation

* Modèle pré-rainé



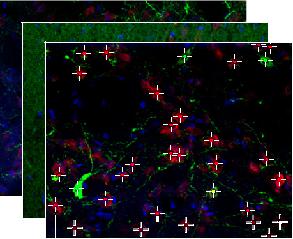
U-Net

Adapté



modèle

**c**



Images avec annotation de ROI

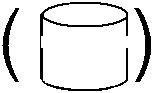
Demande

**d**

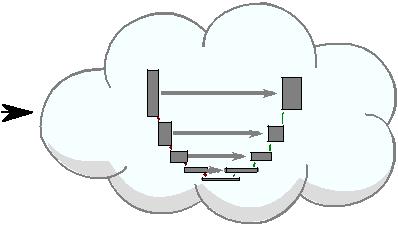


* Préformé

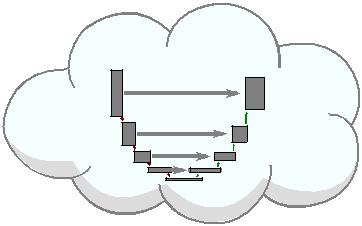
modèle



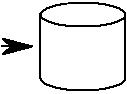
Serveur ou service en nuage



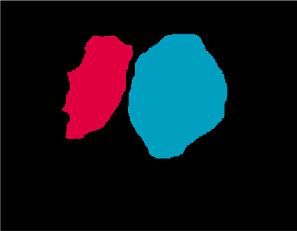
U-Net



Adapté



modèle



Serveur ou service en nuage



Préformé



+ /adapté

modèle

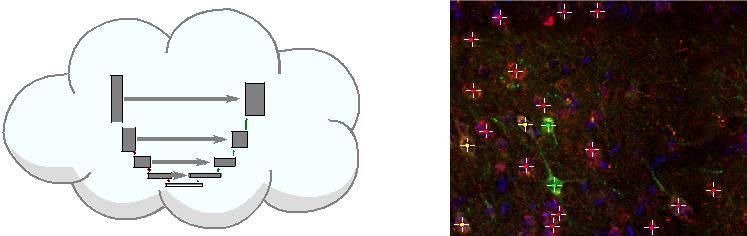
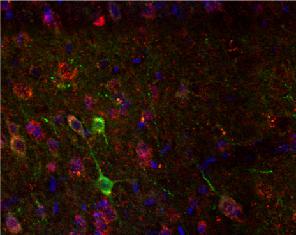
Image d’entrée

U-Net

Segmentation



**e**



Service de serveur ou de nuage



Préformé



+ /adapté U-Net

modèle

|  |  |
| --- | --- |
| Image d’entrée | Détection |

**Fig. 1 | Pipelines du logiciel U-Net.** De gauche à droite : saisir des images et modéliser la formation/l’application réseau (sur la machine locale, un serveur distant dédié ou un service Cloud) généré en sortie. a–c, Adaptation of U-Net to newly annotated data by using transfer learning. **a**, Segmentation with region of interest (ROI) annotations. **b**, Segmentation avec annotations de masque de segmentation. **c**, Détection avec annotations multipoints. **d,e**, Application of the pretrained or adapted U-Net to newly recorded data. **d**, Segmentation. **e**, Detection.

de nouvelles données et de nouvelles tâches par les utilisateurs eux-mêmes, grâce à l’interface familière ImageJ (Fig. 1). Cette capacité permet l’application de U-Net à un large éventail de tâches et le rend accessible à un large éventail de chercheurs qui n’ont pas d’expérience avec l’apprentissage profond. Pour les utilisateurs plus expérimentés, le plugin offre une interface permettant d’adapter des aspects de l’architecture réseau et de se former sur des jeux de données de domaines complètement différents. Le logiciel est livré avec un protocole étape par étape et un tutoriel montrant aux utilisateurs comment annoter les données pour adapter le réseau et décrire les pièges typiques (Note complémentaire 2).

U-Net s’applique aux tâches générales de classification des pixels dans les images planes ou les piles d’images volumétriques avec un ou plusieurs canaux. Tel

les tâches comprennent la détection et le comptage des cellules, c’est-à-dire la prévision d’un point de référence unique par cellule, et la segmentation, c’est-à-dire la délimitation des contours des cellules individuelles. Ces tâches sont un sous-ensemble des tâches de classification les plus répandues, dans lesquelles l’objet d’intérêt est déjà localisé, et seule son étiquette de classe doit être déduite. Bien que l’adaptation à la détection et à la segmentation de structures arbitraires dans les tissus biologiques soit possible compte tenu des données de formation correspondantes, nos expériences se sont concentrées sur les images cellulaires, avec lesquelles la flexibilité de U-Net est montrée dans un large ensemble de tâches de quantification communes, y compris la détection dans les données de fluorescence multicanaux pour les tissus denses, et la segmentation des cellules enregistrées avec des images différentes

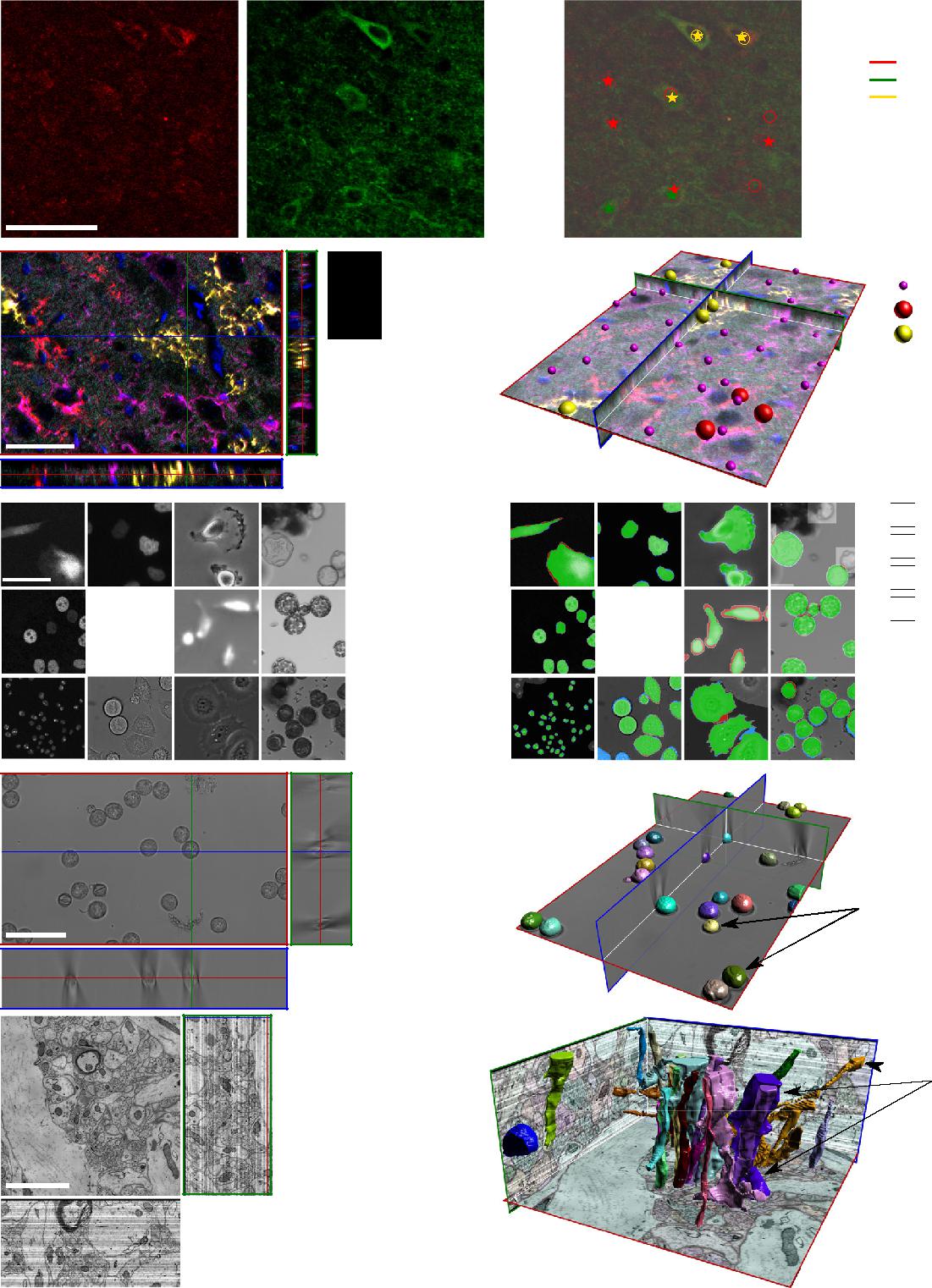
**68** **NATURE METHODS** | VOL 16 | JANVIER 2019 | 67–70 [| www.nature.com/naturemethods](http://www.nature.com/naturemethods)

**MÉTHODES NATURE** BRÈVE COMMUNICATION

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **une tache d’anticorps** | | |  |
| Détection (2D) |  | Analyse de colocalisation |  |  |  |
|  |  |  |  | 50 µm |  |
| (3D) |  | **b** |  |  |  |
|  |  |  |  |
|  | microglialactivité |  |  |  |
| Détection | Quantificationf | 50 µm | |  |
|  |  |  |  |
| Segmentation (2D) | Universalmodelfor | **c** |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| morphometriccelldescription | 50 | µm |  |
|  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cellmorphométrie | données | **d** |  |  |  |  |
|  |  |  |
|  |  |  |  |  |
| image en champ lumineux |  |  | 100 µm | |  |
|  |  |  |  |  |
| (3D) | de |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Segmentation | images | **e** |  |  |  |  |
| Neuritetracing | microscopie électronique |  |  |  |  |
|  |  | 2 | µm |  |
|  |  |  |  |  |
|  | dans |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

expression eYFP



ZICO-1

YFP

PFC

DP

 Expert humain

 U-Net

Anticorps seulement

eYFP seulement

Colocalisation

Détection U-Net

Cellule microgliale (IBA-1)

Cellule étiquetée RFP

Cellule étiquetée YFP

 Correct

 Vérité au sol seulement

 U-Net seulement

 Régions ignorées

 Instances de cellules

 Neurites

**Fig. 2 | Exemples d’applications d’U-Net pour la détection et la segmentation 2D et 3D.** Gauche, données brutes; droite, sortie U-Net (avec comparaison avec l’annotation humaine dans les cas 2D). **a**, Détection de colocalisation dans des images **d’épifluorescence à deux canaux.** b, Détection de cellules microgliales fluorescentes marquées par des protéines dans des piles confocales à cinq canaux. Magenta, toutes les microglies; rouge, vert et cyan : confettis; bleu, tache nucléaire. **c**, segmentation cellulaire en images 2D à partir de fluorescence, contraste d’interférence différentielle, contraste de phase et microscopie à champ lumineux à l’aide d’un modèle commun. **d**, Segmentation des cellules dans les piles d’images à champ lumineux en 3D. **e**, Segmentation de la neurite dans les piles de microscopie électronique.

les modalités en 2D et 3D (Fig. 2 et Note complémentaire 3). Dans tous les cas présentés, U-Net aurait sauvé beaucoup de travail aux scientifiques. Comme le montrent les expériences inter-modalités, la diversité des échantillons biologiques

est trop volumineux pour qu’un seul logiciel puisse tous les couvrir (note supplémentaire 3). En raison de l’approche d’apprentissage, l’applicabilité de U-Net est étendue d’un ensemble de cas spéciaux à un nombre illimité

**NATURE METHODS** | VOL 16 | JANVIER 2019 | 67–70 [| www.nature.com/naturemethods](http://www.nature.com/naturemethods) **69**

BRÈVE COMMUNICATION **MÉTHODES NATURE**

nombre de paramètres expérimentaux. Les résultats de segmentation exceptionnellement précis sur les données volumétriques de champ lumineux, dont l’annotation peut pousser même les experts humains à leurs limites, est une démonstration particulièrement forte des capacités d’un logiciel de quantification automatisé basé sur l’apprentissage profond (Fig. 2d et note supplémentaire 3).

Les critiques insistent souvent sur le besoin de grandes quantités de données de formation pour former un réseau profond. Dans le cas de U-Net, un réseau de base formé sur un ensemble diversifié de données et notre stratégie spéciale d’augmentation des données permettent l’adaptation à de nouvelles tâches avec seulement une ou deux images annotées (Méthodes). Ce n’est que dans des cas particulièrement difficiles que plus de dix images d’entraînement sont nécessaires. Pour juger si un modèle est adéquatement formé, il faut évaluer un ensemble de validation retenu et ajouter successivement des données de formation jusqu’à ce qu’aucune amélioration significative de l’ensemble de validation ne puisse être observée (note supplémentaire 3). Si les ressources informatiques le permettent, la validation croisée avec des fractionnements de train/test attribués aléatoirement évite la sélection d’un ensemble de validation non représentatif au détriment de plusieurs formations. L’effort manuel qui doit être investi pour adapter U-Net à une tâche en cours est généralement beaucoup plus petit que celui nécessaire pour une analyse statistique minimale des données expérimentales. En outre, U-Net offre la possibilité de fonctionner sur des échantillons beaucoup plus grands sans nécessiter d’effort supplémentaire, ce qui le rend particulièrement adapté à la préparation et à l’enregistrement automatisés d’échantillons à grande échelle, Il est probable qu’elles deviendront de plus en plus courantes dans les années à venir.

U-Net a été optimisé pour la convivialité dans les sciences de la vie. L’intégration du logiciel dans ImageJ et un tutoriel étape par étape rendent l’apprentissage en profondeur accessible aux scientifiques n’ayant pas de formation en informatique (vidéos supplémentaires 1 à 4). Il est important de noter que la charge de calcul est praticable pour les laboratoires de sciences de la vie courants (note supplémentaire 1). L’adaptation du réseau à de nouveaux types de cellules ou à de nouvelles modalités d’imagerie va de quelques minutes à quelques heures sur un seul ordinateur équipé d’un GPU grand public. Si le matériel informatique dédié n’est pas disponible en laboratoire, des services cloud communs peuvent être utilisés.

Nos expériences prouvent que l’U-Net produit des résultats de qualité comparable à l’annotation manuelle. Une particularité de U-Net par rapport aux autres outils d’annotation automatique est l’influence individuelle de l’annotateur. Cette fonctionnalité est avantageuse car les chercheurs développent généralement des protocoles individuels dans lesquels plusieurs paramètres sont pris en compte sans être explicitement mentionnés. En raison de leur complexité, ces règles d’étiquetage ne peuvent pas être reproduites par des outils d’étiquetage automatique courants. Cependant, cet avantage peut aussi être un inconvénient : U-Net apprend des exemples fournis. Si les exemples ne sont pas représentatifs de la tâche réelle, ou si l’annotation manuelle dans ces exemples est de mauvaise qualité et incohérente, U-Net échouera à la formation ou reproduira des annotations incohérentes sur de nouvelles données. Cet aspect peut également servir de contrôle de la qualité des annotations manuelles. Dans l’ensemble, U-Net ne peut pas corriger les annotations humaines de faible qualité, mais est un outil permettant d’appliquer des règles d’étiquetage individuelles à de grands ensembles de données et d’économiser ainsi l’effort d’annotation manuelle dans une grande variété de tâches de quantification.

**Contenu en ligne**

Toutes les méthodes, les références supplémentaires, les résumés de rapports de Nature Research, les données sources, les énoncés de disponibilité des données et les codes d’accès connexes sont disponibles à [https://doi.org/10.1038/](https://doi.org/10.1038/s41592-018-0261-2) [s41592-018-0261-2](https://doi.org/10.1038/s41592-018-0261-2).

Reçu : 26 juillet 2018; Accepté : 19 novembre 2018;

Publié en ligne : 17 décembre 2018

**Références**

1. Sommer, C, Strähle, C, Koethe, U. & Hamprecht, F. A. in Ilastik : interactive learning and segmentation toolkit in *IEEE Int. Symp. Biomed. Imaging.* 230–233 (IEEE : Piscataway, NJ, USA, 2011).

2. Arganda-Carreras, I. et al. *Bioinformatics* **33**, 2424-2426 (2017).

3. Ronneberger, O., Fischer, P. & Brox, T. U-Net : réseaux convolutionnels pour la segmentation d’images biomédicales. in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015* Vol. 9351, 234-241 (Springer, Cham, Suisse, 2015).

1. Rusk, N. *Nat. Méthodes* **13**, 35 (2016).
2. Webb, S. *Nature* **554**, 555-557 (2018).

6. Sadanandan, S. K., Ranefall, P., Le Guyader, S. et Wählby, C. *Sci. Rep.* **7**, 7860 (2017).

7. Weigert, M. et al. *Nat. Méthodes*<https://doi.org/10.1038/s41592-018-0216-7> (2018).

8. Haberl, M. G. et al. *Nat. Methods* **15**, 677–680 (2018). 9. Ulman, V. et al. *Nat. Méthodes* **14**, 1141–1152 (2017). 10. Schneider, C. A., Rasband, W. S. et Eliceiri, K. W. *Nat. Méthodes* **9,**

671–675 (2012).

**Remerciements**

Ces travaux ont été soutenus par le ministère fédéral allemand de l’éducation et de la recherche (BMBF) dans le cadre du projet MICROSYSTEMS (0316185B) à T.F. et A.D.; le Prix Bernstein 2012 (01GQ2301) à I.D.; le ministère fédéral de l’économie et de l’énergie (ZF4184101CR5) à A.B.; la Deutsche Forschungsgemeinschaft (DFG) par l’intermédiaire du centre de recherche collaborative KIDGEM (SFB 1140) à D.M., Ö.Ç., T.F. et O.R., et (SFB 746, INST 39/839,840,841) à K.P.; les Clusters of Excellence BIOSS (EXC 294) à T.F., D.M., R.B., A.A., Y.M., D.S., T.L.T., M.P., K.P., M.S., T.B. et O.R.; BrainLinks-Brain-Tools (EXC 1086) à Z.J., K.S., I.D. et T.B.; accorde l’IR 1908/3-1 à J.D., l’IR 1908/6-1 à Z.J. et K.S., et l’IR 1908/7-1 à I.D.; la Fondation nationale suisse de la science (subvention 173880 du FNS) aux AA; le CER Starting grant OptoMotorPath (338041) à I.D.; et le réseau d’excellence FENS-Kavli (FKNE) à I.D. Nous remercions F. Prósper, E. Bártová, V. Ulman, D. Svoboda, G. van Cappellen, S. Kumar, T. Becker et le consortium Mitocheck pour avoir fourni une riche diversité de jeux de données à travers le défi de segmentation ISBI. Nous remercions P. Fischer pour les annotations d’images manuelles. Nous remercions S. Wrobel pour la préparation des microspores du tabac.

**Contributions des auteurs**

T.F., D.M., R.B., Y.M., Ö.Ç., T.B. et O.R. ont sélectionné et conçu les expériences de calcul. T.F., R.B., D.M., Y.M., A.B. et Ö.Ç. ont effectué les expériences suivantes : R.B., D.M., Y.M. et A.B. (2D), et T.F. et Ö.Ç. (3D). R.B., Ö.Ç., A.A., T.F. et O.R. ont implémenté les extensions U-Net dans caffe. T.F. a conçu et implémenté le plugin Fidji. D.S. et M.S. ont sélectionné, préparé et enregistré l’ensemble de données sur le kératinocyte PC3-HKPV. T.F. et O.R. ont préparé l’ensemble de données sur le pollen en suspension dans l’air BF1-POL. A.D., S.W., O.T., C.D.B. et K.P. ont sélectionné, préparé et enregistré les ensembles de données de protoplastes et de microspores BF2-PPL et BF3-MiSp. T.L.T. et M.P. ont préparé, enregistré et annoté les données pour l’expérience de prolifération microgliale. J.D., K.S. et Z.J. ont sélectionné, préparé et enregistré l’ensemble de données optogénétiques. I.D., J.D. et Z.J. ont annoté manuellement l’ensemble de données optogénétiques. I.D., T.F., D.M., R.B., Ö.Ç., T.B. et O.R. ont écrit le manuscrit.

**Competing interests**

The authors declare no competing interests.

**Additional information**

**Supplementary information** is available for this paper at [https://doi.org/10.1038/](https://doi.org/10.1038/s41592-018-0261-2) [s41592-018-0261-2](https://doi.org/10.1038/s41592-018-0261-2).

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Correspondence and requests for materials** should be addressed to O.R.

**Publisher’s note:** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2018

**70** **NATURE METHODS** | VOL 16 | JANUARY 2019 | 67–70 | [www.nature.com/naturemethods](http://www.nature.com/naturemethods)

**NATURE METHODS** BRIEF COMMUNICATION

**Methods**

**Network architecture.** U-Net is an encoder–decoder-style neural network that solves semantic segmentation tasks end to end; it extends the fully convolutional network from Long et al.11 (Supplementary Fig. 1). Its 26 hidden layers can be logically grouped in two parts: (i) an encoder that takes an image tile as input and successively computes feature maps at multiple scales and abstraction levels thus yielding a multilevel, multiresolution feature representation, and (ii) a decoder that takes the feature representation and classifies all pixels/voxels at original image resolution in parallel. Layers in the decoder gradually synthesize the segmentation, starting at low-resolution feature maps (describing large-scale structures) and moving to full-resolution feature maps (describing fine-scale structures).

The encoder is a vgg-style convolutional neural network12. It consists of the repeated application of two convolutions (without padding), each followed by a leaky rectified linear unit (ReLU) with a leakage factor of 0.1 and a max -pooling operation with stride two halving the resolution of the resulting feature map.

Convolutions directly following down-sampling steps double the number of feature channels.

The decoder consists of repeated application of an up-convolution (an up-convolution is equivalent to a bed-of-nails up-sampling by a factor of two followed by a convolution with an edge length of two) that halves the number of feature channels, then concatenation with the cropped encoder feature map at corresponding resolution and two convolutions, each followed by leaky ReLUs with a leakage factor of 0.1.

At the final layer, a 1 ×​1 convolution is used to map feature vectors to the desired number of classes *K.*

For training, the final *K*-class scores are transformed to pseudoprobabilities with soft-max before they are compared to the ground-truth annotations in one-hot encoding using cross-entropy.

If not explicitly mentioned, operations are isotropic, and convolutions use a kernel edge length of three pixels. We use only the valid part of each convolution; i.e., the segmentation map contains only the pixels for which the full context is available in the input image tile.

This fully convolutional network architecture allows users to freely choose the input-tile shape, with the restriction that the edge lengths of feature maps before max-pooling operations must be even. Thus, arbitrarily large images can be processed by using an overlap-tile strategy in which the stride is given by the output-tile shape. To predict pixels in the border region of an image, missing context is extrapolated by mirroring the input image.

The ImageJ plugin presents only valid tile shapes to the user. For 2D networks, the user can also let the plugin automatically choose an optimal tiling according to the available GPU memory. For this process, the amount of memory used

by the given architecture for different tile sizes must be measured on the client PC beforehand, and corresponding information must be stored to the model definition file by using the provided MATLAB script caffe-unet/matlab/unet/ measureGPUMem.m.

*3D U-Net.* The 3D U-Net architecture for volumetric inputs includes only a slight modification of its 2D counterpart. First, input tiles have a shape of 236 × 236 ×​100 voxels. Second, owing to memory limitations, we halve the number of output features of each convolution layer except for the last up-convolution step, which produces only 32 channels. Third, to address anisotropic image voxel sizes, convolutions and pooling at the finest-resolution levels are 2D until the voxel extents are approximately isotropic. At all remaining resolution levels, operations are applied along all three dimensions.

**Weighted soft-max cross-entropy loss.** We use pixel-weighted soft-max cross-entropy loss to enable changing the influence of imbalanced classes in semantic segmentation. The loss is computed as

|  |  |  |  |
| --- | --- | --- | --- |
| *l (I* ) : = w (*x****) log*** | exp (*ŷy(****x) (x))*** | |  |
| *K* | exp (*ŷ (****x))*** |  |
| ***x*** Ω | *k*=0 | *k* |  |

where *x* is a pixel in image domain Ω, *ŷk* : Ω→R is the predicted score for class

*k* {0,…,*K}, K* is the number of classes, and *y*:Ω→​{0,…,*K*} is the ground-truth segmentation map. Thus, *ŷy* ***(x) (x)*** is the predicted score for ground-truth class *y(****x***) at position ***x***. As defined above, *w* : Ω→R≥0 is the pixelwise-loss-weight map.

We use loss weighting to optimize class balancing and handle regions without annotations (termed ‘ignored regions’ below), by using weight map *w*bal. We additionally enforce instance separation, but using weight *w*sep, as described in the following sections. The final loss weights are then given by

*w* : = *w*bal + *λw*sep

where *λ* ℝ≥0 controls the importance of instance separation.

*Class balancing and regions with unknown ground truth.* In our images, most pixels belong to the background class, and in many cases, this class is homogeneous and easy to learn. Therefore, we decrease the weight of background pixels by the factor

*v*bal [0,1],​ as compared with foreground pixels, thus resulting in the class-balancing weight function

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1 | *y (****x***) > 0 |  |
|  |  |  |
|  |  |  |  |  |
| *wbal* ***(x***) : = |  |  | *y (****x*** ) = 0 |  |
| *v* bal | |  |
|  |  | 0 | *y (****x***) unknown (i. e. , ignored regions) |  |
|  |  |  |

We observed slightly better segmentation results when replacing the step-shaped cutoff at the edges of foreground objects by a smoothly decreasing Gaussian function for the weighted loss computation; therefore, we define

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *w′bal* ***(x***) := | |  |  |  |  |
|  | 1 |  |  | *y (****x***) > 0 |  |
|  |  |  |  |
|  |  |  |  |  |  |
|  |  | 2 |  |  |  |
|  |  |  |
|  |  | *d*1 ***(x)*** | |  |  |
|  | *v* bal + (1−*vbal*) exp − |  |  | *y (* ***x***) = 0 |  |
|  | 2 |  |  |
|  |  |  |  |
|  |  | 2*σ*bal |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | 0 |  |  | *y (****x***)inconnu (p. ex., |  |
|  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  | ignored regions) |  |
|  |  |  |  |  |
|  |  |  |  |  |  |

where *d*1***(x***) is the distance to the closest foreground object, and *σ*bal is the s.d. of the Gaussian.

**Detection.** The detection tasks are reformulated as segmentation tasks by rendering a segmentation map containing a small disk (2D) or a small ball (3D) at each annotated location. Each disk or ball is surrounded by a ring-shaped or spherical-shell-shaped ignored region with outer radius *r*ign, in which both *wbal* and *w'bal* are set to zero. These regions help stabilize the detection process in case of inaccurate annotations but are not essential to the approach.

*Instance segmentation.* Semantic segmentation classifies the pixels/voxels of an image. Therefore, touching objects of the same class will end up in one joint segment. Most often, users want to measure the properties of individual object instances (for example, cells). Therefore, the pure semantic segmentation must be turned into an instance-aware semantic segmentation. To do so, we insert an artificial 1-pixel-wide background ridge between touching instances in the ground-truth segmentation mask (Supplementary Fig. 2a). To force the network to learn these ridges, we increase their weight in the loss computation, such that the thinnest ridges have the highest weights. We approximate the ridge width at each pixel by the sum of the distance *d*1 to its nearest instance and the distance *d*2 to its second-nearest instance. From this, we compute the weight map as

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | *(d*1 ***(x*** ) + *d2****(x))*** | 2 |  |  |
|  |  |  |  |  |
|  | − |  |  |  |  |
|  |  |  |  |
| *w*sep ***(x***) : = exp | 2 |  |  |
|  |  |  |  |  |
|  |  | 2*σ*sep |  |  |  |

which decreases following a Gaussian curve with s.d. *σ*sep (Supplementary Fig. 2b).

**Tile sampling and augmentation.** Data augmentation is essential to teach the expected appearance variation to the neural network with only a few annotated images. To become robust to translations and to focus on relevant regions in the images, we first randomly draw spatial locations of image tiles that are presented to the network from a user-defined probability density function. For all our experiments, we used the normalized weight map *wbal* to sample the spatial location; i.e., foreground objects are presented to the network ten times more often (according to our selection of *v*bal =​0.1) than background regions. Tiles centered around an ignored region are never selected during training. We then draw a random rotation angle (around the optical axis in 3D) from a uniform distribution in a user-defined range. Finally, we generate a smooth deformation field by placing random displacement vectors with user-defined s.d. of the magnitude for each component on a very coarse grid. These displacement vectors are used to generate a dense full-resolution deformation field by using bicubic interpolation (Supplementary Fig. 3). Rigid transformations and elastic deformations are concatenated to look-up intensities in the original image during tile generation.

We additionally apply a smooth strictly increasing intensity transformation to become robust to brightness and contrast changes. The intensity mapping curve is generated from a user-defined number of equidistant control points in the normalized [0,1] source intensity range. Target intensities at the control points are drawn from uniform distributions with user-defined ranges. The sampling process enforces increased intensities at subsequent control points, and spline interpolation between the control points ensures smoothness. All data augmentation is applied on the fly to the input image, label map and weight map during network training13.

**Training.** All networks were trained on an nVidia TITAN X with 12 GB GDDR5 RAM, by using cuda 8 and cuDNN 6 with caffe14 after our proposed extensions were applied. The initial network parameters were drawn from a Gaussian distribution with s.d. *σ* = 2∕*n*in , where *n*in is the number of inputs of one neuron



**NATURE METHODS** | [www.nature.com/naturemethods](http://www.nature.com/naturemethods)

BRIEF COMMUNICATION **NATURE METHODS**

of the respective layer15. For all experiments, the raw image intensities per channel were normalized to the [0,1] range before training, by using

* : =*I* min{*I* } max{*I* } min{*I }*

where *I* is the raw intensity, and Î is the normalized intensity.

**Transfer learning.** Adaptation of a pretrained U-Net to a new dataset by using annotated data is called transfer learning or fine-tuning. Transfer learning leverages the knowledge about the different cell datasets already learned by the U-Net and usually requires considerably fewer annotated data and training iterations than training from scratch.

Transfer to a new dataset is based on the same training protocol as described above. The user must provide only raw images and corresponding annotations as ImageJ ROIs or pixel masks (detection, one multipoint ROI per class and image; segmentation, one regional ROI per object or pixel masks) (Supplementary Note 2). The plugin performs image rescaling, pixel-weight *w(****x***) generation and parametrization of the caffe training software.

**Evaluation metrics.** *Object intersection over union.* The intersection over union (IoU) measures how well a predicted segmentation matches the corresponding ground-truth annotation by dividing the intersection of two segments by their union. Let S : = {*s* 1*, …, s****N*** } be a set of *N* pixels. Let G S be the set of pixels belonging to a ground-truth segment and P S be the set of pixels belonging to the corresponding predicted segment. The IoU is defined as

|  |  |  |
| --- | --- | --- |
| *M*IoU ( G, P) : = | G P |  |
| G P |  |

IoU is a widely used measure, for example, in the Pascal VOC challenge16 or

the ISBI cell-tracking challenge17 . *M*IoU [0,1],​ with 0 meaning no overlap and 1 meaning a perfect match. In our experiments, a value of 0​.7 indicates a good segmentation result, and a value of 0​.9 is close to human annotation accuracy.

We first determine the predicted objects by performing connected component labeling with eight-neighborhood on the binary output of the U-Net. This procedure yields candidate segments P*i* . Then we compute the IoU for every pair of output and ground-truth segments G*j : M*IoU (G *j*, P*i*) and apply the Hungarian algorithm on *M*IoU to obtain 1:1 correspondences maximizing the average IoU. Unmatched segments or matches with zero IoU are considered false positives and false negatives, respectively. The average IoU is computed on the basis of the ground-truth annotations; i.e., false negatives contribute to the average IoU with a value of 0. False positives, however, are not considered in the average IoU. We provide a detailed analysis of false-positive segmentations produced by U-Net (Supplementary Note 3).

*F measure.* We measure the quality of a detector by using the balanced *F*1 score (denoted *F* measure), which is the harmonic mean of precision and recall. Let *G* be the set of true object locations and *P* be the set of predicted object locations. Then precision and recall are defined as

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *M*Precision : = | *G* P | and*M*Recall : = | | *G* P | . |  |
| *P* | *G* |  |
| The *F* measure is then given by | |  |  |  |  |  |
|  | *F*1 : 2 | *M*Precision *M*Recall | |  |  |  |
|  |  | *M*Precision + *M*Recall |  |  |  |  |

Pixel-accurate object localization is nearly impossible to reach and is rarely necessary in biomedical applications; therefore, we introduce a tolerance *d*match for matching detections and annotations. To obtain one-to-one correspondences, we first compute the pairwise distances of ground-truth positions to detections. Predictions with distance greater than *d*match to any ground-truth position

can be directly classified as false positives. Similarly, ground-truth positions without detections in the *d*match range are classified as false negatives. We apply the Hungarian algorithm to the remaining points to obtain an assignment of predictions to ground-truth positions minimizing the total sum of distances

of matched positions. Correspondences with distance greater than *d*match and unmatched positions in either ground-truth or prediction are treated as false-negative or false-positive detections as before.

**Statistics.** In the microglial detection experiment, significant differences between random clone distribution and clone clustering are reported. We compared the actual measurements obtained through U-Net detection with a Monte Carlo experiment simulating the null hypothesis of equally distributed confetti-marked microglia. We report the ninety-eighth-percentile range of 10,000 simulation runs (bootstrap iterations) leading to a two-tailed nonparametric test. If the measurement is outside the ninety-eighth-percentile, the null hypothesis does not hold, with a *P* value <0.02.

**Code availability.** We have provided prebuilt binary versions of the U-Net caffe extensions for Ubuntu Linux 16.04 at [https://lmb.informatik.uni-freiburg.de/](https://lmb.informatik.uni-freiburg.de/resources/opensource/unet/) [resources/opensource/unet/](https://lmb.informatik.uni-freiburg.de/resources/opensource/unet/). Our changes to the source code of the publicly available caffe deep-learning framework14 are additionally provided ([https://](https://github.com/BVLC/caffe/) [github.com/BVLC/caffe/](https://github.com/BVLC/caffe/)) as a patch file with detailed instructions on how

to apply the patch and build our caffe variant from the source on the project website and within the plugin’s Help function. Binary installation requires only unpacking the archive and installing the required third-party libraries, as can be done within several minutes on an Ubuntu 16.04 machine, depending on the internet connection for fetching the packages. Building from scratch requires installing the dependent development libraries, checking out the given tagged version of the BVLC caffe master branch and applying the patch. With a normal internet connection, package installation requires a few minutes. Cloning the BVLC master repository requires less than a minute, and applying the patch imposes no measurable overhead. Configuring and building the package requires approximately 10 to 15 min. The U-Net segmentation plugin for Fiji/ImageJ is available at <http://sites.imagej.net/Falk/plugins/>or through the ImageJ updater within Fiji. Source code is included in the plugin jar File Unet\_Segmentation. jar. Installation with the Fiji updater requires only a few seconds. The trained caffe-models for the 2D and 3D U-Net are available at <https://lmb.informatik.uni-freiburg.de/resources/opensource/unet/>.

**Reporting Summary.** Further information on research design is available in the Nature Research Reporting Summary linked to this article.

**Data availability**

Datasets F1-MSC, F2-GOWT1, F3-SIM, F4-HeLa, DIC1- HeLa, PC1-U373 and PC2-PSC are from the ISBI Cell Tracking Challenge 2015 (ref. 17). Information on how to obtain the data can be found at [http://celltrackingchallenge.net/datasets.](http://celltrackingchallenge.net/datasets.html) [html](http://celltrackingchallenge.net/datasets.html), and free registration for the challenge is currently required. Datasets PC3-HKPV, BF1-POL, BF2-PPL and BF3-MiSp are custom and are available from the corresponding author upon reasonable request. Datasets for the detection experiments partially contain unpublished sample-preparation protocols and are currently not freely available. After protocol publication, datasets will be made available on an as-requested basis. Details on sample preparation for our life science experiments can be found in Supplementary Note 3 and the Life Sciences Reporting Summary.

**References**

\11.\Long, J., Shelhamer, E. & Darrell, T. Fully convolutional networks for semantic segmentation. in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)* 3431–3440 (IEEE, Piscataway, NJ, USA, 2015).

12. Simonyan, K. et Zisserman, A. Preprint à<https://arxiv.org/abs/1409.1556> (2014)

\13.\Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T. & Ronneberger, O. 3D U-Net: learning dense volumetric segmentation from sparse annotation. in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2016* Vol. 9901, 424–432 (Springer, Cham, Switzerland, 2016).

14. Jia, Y. et al. Preprint at<https://arxiv.org/abs/1408.5093>(2014). \15.\He, K., Zhang, X., Ren, S. & Sun, J. Preprint at [https://arxiv.org/](https://arxiv.org/abs/1502.01852)

[abs/1502.01852](https://arxiv.org/abs/1502.01852) (2015).

\16.\Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. & Zisserman, A.

*Int. J. Comput. Vis.* **88**, 303–338 (2010).

17. Maška, M. et al. *Bioinformatics* **30**, 1609-1617 (2014).

**NATURE METHODS** | [www.nature.com/naturemethods](http://www.nature.com/naturemethods)



Corresponding author(s): Olaf Ronneberger

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistical parameters

When statistical analyses are reported, confirm that the following items are present in the relevant location (e.g. figure legend, table legend, main text, or Methods section).

n/a Confirmed

The exact sample size *(n*) for each experimental group/condition, given as a discrete number and unit of measurement



An indication of whether measurements were taken from distinct samples or whether the same sample was measured repeatedly



The statistical test(s) used AND whether they are one- or two-sided



*Only common tests should be described solely by name; describe more complex techniques in the Methods section.*

A description of all covariates tested



A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons



A full description of the statistics including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)



For null hypothesis testing, the test statistic (e.g. *F, t, r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted



*Give P values as exact values whenever suitable.*

For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings



For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes



Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated



Clearly defined error bars



*State explicitly what error bars represent (e.g. SD, SE, CI)*

*Our web collection on statistics for biologists may be useful.*

|  |
| --- |
| nature research | reporting summary |



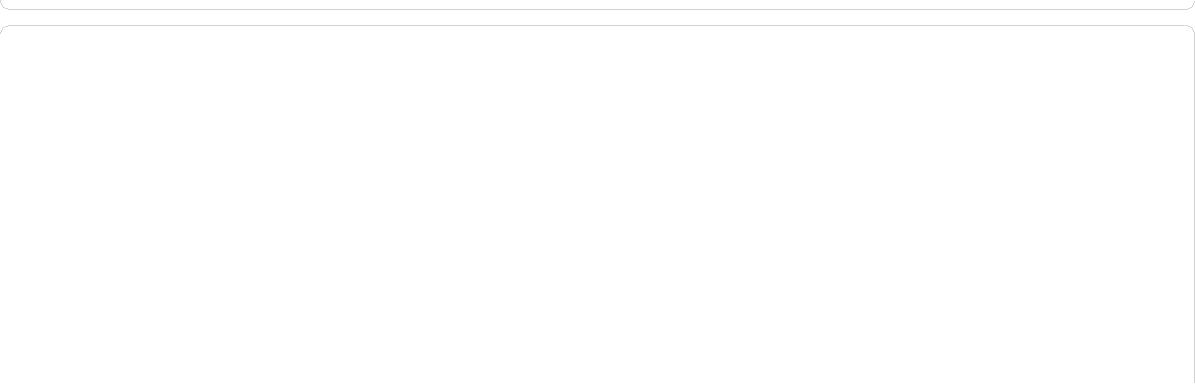
Software and code

Policy information about availability of computer code



Data collection Optogenetics experiment (2D detection): ZEN2010 B SP1 Version 6.0.0.485; Microglia experiment (3D detection): Olympus FluoView 1000; Cell segmentation: BF2-PPL,. BF3-MiSp: LA (Live Acquisition) Software, TILL Photonics GmbH, Germany, Version 2.2.1.0, PC3-HKPV: ZEISS AxioVision

All other datasets are thirdparty.



|  |  |  |
| --- | --- | --- |
| Data analysis | caffe\_unet (custom, binaries and source code provided as Supplementary Software 1-4). More recent versions can be found at https:// | |
|  | lmb.informatik.uni-freiburg.de/resources/opensource/unet/ | |
|  | caffe\_unet is a variant of the open source deep learning framework caffe (https://github.com/BVLC/caffe). It is the core of this study. Our | |
|  | U-Net segmentation plugin for Fiji (ImageJ) is a front-end that interfaces caffe from ImageJ to provide easy integration of segmentation | |
|  | and detection using deep learning into Fiji protocols for non-computer scientists. | |
|  | U-Net Segmentation plugin (custom, provided via the Fiji community repository sites.imagej.net): version at time of submission: https:// | |
|  | sites.imagej.net/Falk/plugins/Unet\_Segmentation.jar-20181112152803 | |
|  | The plugin is maintained to work with the most recent Fiji version and requires Java 8. Future updates may break backwards- | |
|  | compatibility if the structure of the ImageJ API changes. | |
|  | Fiji (ImageJ) (https://fiji.sc). Version at time of submission: 1.52h. | |
|  | Fiji is a variant of the open source image processing software ImageJ with many useful extensions for image analysis in the life sciences. | |
|  |  |  |
|  |  |  |



|  |
| --- |
| *April 2018* |

1

No statistical methods were used to predetermine sample sizes in the microglia experiment. We ensured that they were similar to those generally employed in the field. We chose a random 25% training / 75% test data split to provide a sufficient amount of training data to the network while still having enough data in the test set to draw statistically significant conclusions. Further experiments with other splits were not performed due to limited computational resources.

MATLAB (https://de.mathworks.com/products/matlab.html): version used in this study: R2016a  MATLAB is a commercial general purpose suite optimized for fast and versatile numeric operations on multi-dimensional data. It provides a wide range of toolboxes for various tasks, including image processing. Many of the experiments for this study were performed with MATLAB as frontend, but it is not required to reproduce any of our findings.



For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers upon request. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

* Accession codes, unique identifiers, or web links for publicly available datasets
* A list of figures that have associated raw data
* A description of any restrictions on data availability



The datasets F1-MSC, F2-GOWT1, F3-SIM, F4-HeLa, DIC1-HeLa, PC1-U373, and PC2-PSC are from the ISBI Cell Tracking Challenge 2015. Informations on how to obtain the data can be found at http://celltrackingchallenge.net/datasets.html and at time of publication required free-of-charge registration for the challenge.

The datasets PC3-HKPV, BF1-POL, BF2-PPL, and BF3-MiSp are custom and are available from the corresponding author upon reasonable request.

Datasets for the detection experiments partially contain unpublished sample preparation protocols, and are currently not freely available. Upon protocol publication datasets will be made available on request-basis.



|  |
| --- |
| nature research | reporting summary |



Field-specific reporting

Please select the best fit for your research. If you are not sure, read the appropriate sections before making your selection.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Life sciences |  | Behavioural & social sciences |  | Ecological, evolutionary & environmental sciences |

For a reference copy of the document with all sections, see nature.com/authors/policies/ReportingSummary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.



Sample size

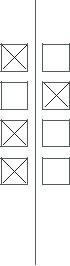


|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Data exclusions |  | All available annotated images were included without selection. The subset of raw images for annotation was randomly chosen from all | |  |  |  |  |
|  |  | available images and randomly split into training and test sets. A validation set was not required, because we did not optimize | |  |  |  |  |
|  |  | hyperparameters of our model besides the number of training iterations, which we selected so that we obtained the best average results on | |  |  |  |  |
|  |  | the test set, but the full curves are given for reference and show that the number of iterations is not a critical parameter. | |  |  |  |  |
| Replication |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | Facial nerve transection (microglia experiment) was performed in at least three separate occasions and cohorts were randomly assigned to all | |  |  |  |  |
|  |  |  |  |  |
|  |  | test groups (i.e., time after injury). All replication attempts were successful. | |  |  |  |  |
|  |  | We repeated the finetuning experiments for 2D cell segmentation on randomly chosen subsets of the training data. If the training set is | |  |  |  |  |
|  |  | sufficiently large (100 cells or more) the prediction variance tends towards zero showing high reproducibilityreproducability.. | |  |  |  |  |
| Randomization |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | Sample allocation was random in all stages of the experiments. | |  |  |  |  |
|  |  |  |  |  |
|  |  | A coworker of TLT randomly assigned numbers to mice, slides and images for data acquisition and processing (microglia experiment). | |  |  |  |  |
| Blinding |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | Only in the microglia experiment different groups were analyzed and compared. The experiment was performed in a blinded manner due to | |  |  |  |  |
|  |  |  |  |  |
|  |  | random assignment of numbers to mice, slides and images for data acquisition and processing. Final results were reassigned to their | |  |  |  |  |
|  |  | respective test groups by TLT and DM. | |  |  |  |  |
|  |  |  |  |  |  | *April* |  |
| Reporting for specific materials, systems and methods | | | |  |  |  |
|  |  | *2018* |  |
|  |  |  |  |  | |  |



2

|  |  |  |  |
| --- | --- | --- | --- |
| Materials & experimental systems |  | Methods | |
|  |  |  |  |
| n/a Involved in the study |  | n/a | Involved in the study |
| Unique biological materials |  |  | ChIP-seq |
| Antibodies |  |  | Flow cytometry |
| Eukaryotic cell lines |  |  | MRI-based neuroimaging |
|  |  |  |  |



Palaeontology

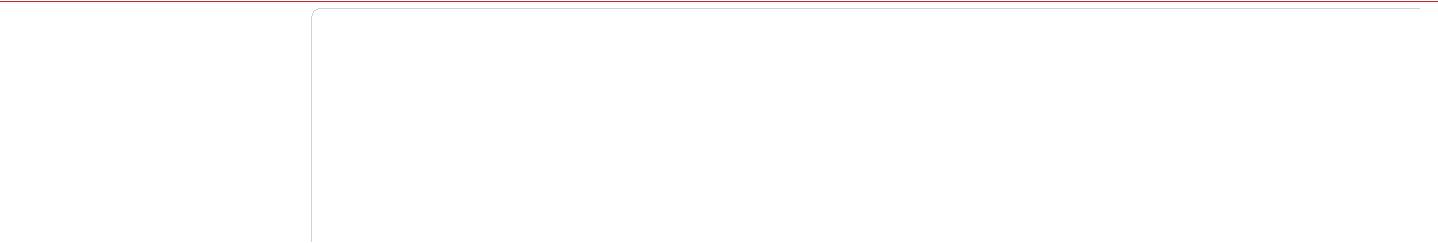


Animals and other organisms



Human research participants

Antibodies



|  |
| --- |
| nature research | reporting |



Antibodies used

Validation

Optogenetics experiment: We used antibodies against NeuN 1:500 (EMD Millipore Corp., Catalog Nr.: ABN78, polyclonal

produced in rabbit, Lot: 2951836), CaMKIIα (A-1, Santa Cruz Biotechnology, Catalog Nr.: Sc-13141, monoclonal produced in mouse, LOT: B1712) and Parvalbumin (Sigma-Aldrich, Catalog Nr.: P3088, clone PARV-19, monoclonal produced in mouse, LOT: 122M4774V) as primary antibodies. As secondary antibodies we used Goat Anti-Rabbit IgG (H+L), Alexa Fluor® 647 conjugate (Sigma-Aldrich, Catalog Nr.: AP187SA6, polyclonal, LOT: 2277868) and Donkey Anti-Mouse IgG, Alexa Fluor® 647 conjugate (Sigma-Aldrich, Catalog Nr.: AP192SA6, polyclonal, LOT: 2613129).

Microglia experiment: 1:500 rabbit anti-Iba-1 (Wako 019-19741); 1:1,000 donkey anti-rabbit IgG (H+L) Highly Cross-Adsorbed Secondary Antibody, Alexa Fluor 647 (Life Technologies A-31573)



Optogenetics experiment: Validation of the antibodies was done by immunoelectrophoresis and/or ELISA by the manufacturer to ensure specificity and minimal cross-reaction with rat serum proteins and/or extensive use in various publications.



NeuN:

Immunocytochemistry Analysis: A 1:500 dilution from a representative lot detected NeuN in cultured embryonic E18 rat hippocampus neurons. [EMD Millipore Corp.]

Immunohistochemistry Analysis: A 1:500 dilution from a representative lot detected NeuN-positive neurons in human (cerebellum and cerebral cortex) and mouse (hippocampus) brain tissue sections. [EMD Millipore Corp.]

A representative lots immunostained 4% paraformaldehyde-fixed neurons isolated from snail brain (Nikolić, L., et al. (2013). J.

Exp. Biol. 216 (Pt 18):3531-3541).

Representative lots detected NeuN-positive neurons in frozen brain sections from 4% paraformaldehyde-perfused mice (Cherry, J.D., et al. (2015). J. Neuroinflammation. 12:203; Huang, B. et coll. (2015). Neuron. 85(6):1212-1226; Ataka, K. et coll. (2013). PLoS One. 8(11):e81744).

A representative lots detected NeuN-positive neurons in formalin-fixed and paraffin-embedded amyotrophic lateral sclerosis (ALS) human brain tissue sections (Fratta, P., et al. (2015). Neurobiol. Aging. 36(1):546.e1-7).

A representative lot detected NeuN-positive neurons in rat brain tissue sections (Mendonça, M.C., et al. (2013).Toxins.

5(12):2572-2588).

Representative lot detected NeuN in mouse and rat brain tissue lysates (Huang, B., et al. (2015). Neuron. 85(6):1212-1226; Mendonça, M.C., et coll. (2013).Toxins. 5(12):2572-2588).

CamKII:

Maurya, SK. et al. 2018. Cell Rep. 24: 2919-2931. ; Lee, Y. et al. 2018. Mol. Neurobiol. 55: 5658-5671. ; Go, J. et al. 2018. Int. J.

Mol. Med. 42: 1875-1884. ; Oliver, RJ. et al. 2017. Genes Brain Behav. ; Zhang, Y. et coll. 2017. J. Neurosci. 37 : 9741-9758;

Raffeiner, P. et al. 2017. Oncotarget. 8 : 3327-3343. ; Yan, X. et al. 2016. PLoS ONE. 11: e0162784. ; Deng, H. et al. 2016. J.

Neurosci. 36 : 7580-8. ; Volle, J. et al. 2016. Cell Rep. 15: 2400-10. ; Kim, H. et al. 2016. Mol. Neurobiol. ; Sheng, L. et al. 2015. J.

Neurosci. 35 : 1739-52. ; Cipolletta, E. et al. 2015. PloS one. 10: e0130477. ; Gratuze, M. et al. 2015. Hum. Mol. Genet. 24:

86-99. ; Wang, S. et al. 2015. J Biomed Res. 29: 370-9. ; Okada, R. et al. 2015. Neuroscience. 299 : 134-45. ; Okada, R. et al. 2014.

Neurochem. Res. 39: 875-82. ; Mizuki, D. et al. 2014. J Ethnopharmacol. 156 : 16-25. ; Mizuki, D. et al. 2014. J. Pharmacol. Sci.

1. 457-67. ; Kim, H. et al. 2014. Mol Brain. 7: 77. ; Le, XT. et al. 2013. Neurochem. Res. 38: 2201-15. ; Cao, C. et al. 2013. PLoS biology. 11: e1001478. ; Yen, YH. et al. 2011. Mol Cell Proteomics. -. ; Zhao, Q. et al. 2010. J Ethnopharmacol. 131 : 377-385. ; España, J. et al. 2010. Biol. Psychiatry. 67: 513-521. ; Zhang, G.R. et coll. 2009. Neuroscience. 159 : 308-315. ; Cheng, HH. et al. 2009. J. Neurosci. Res. 87: 460-469. ; Chang, CW. et al. 2007. Proteomics Clin Appl. 1: 1499-1512. ; Saha, S. et al. 2007. Brain Res. 1148: 38-42. ; Saha, S. et al. 2006. Biochem. Biophys. Res. Commun. 350: 444-449.

PV:

Recognizes parvalbumin in a Ca2+ ion-dependent manner. Does not react with other members of the EF-hand family such as calmodulin, intestinal calcium-binding protein, S100A2 (S100L), S100A6 (calcyclin), the α chain of S-100 (i.e. in S-100a and S-100ao), or the β chain (i.e. in S-100a and S-100b). [Sigma-Aldrich]

Goat anti rabbit IgG Alexa Fluor 647:

Based on immunoelectrophoresis and/or ELISA, the antibody reacts with whole molecule rabbit IgG. It also reacts with the light chains of other rabbit immunoglobulins. No antibody was detected against non-immunoglobulin serum proteins.

The antibody has been tested by ELISA and/or solid-phase absorbed to ensure minimal cross-reaction with human, mouse, and rat serum proteins, but may cross-react with immunoglobulins from other species. [Sigma-Aldrich]

Donkey anti mouse IgG Alexa Fluor 647:

Based on immunoelectrophoresis and/or ELISA, the antibody reacts with whole molecule mouse IgG. It also reacts with the light chains of other mouse immunoglobulins. No antibody was detected against non-immunoglobulin serum proteins.

|  |
| --- |
| summary |

|  |
| --- |
| *April 2018* |

3

Microglia experiment: female CX3CR1-creER/wt ; R26R-Confetti/wt (in C57/B6J background) were bred in specific-pathogen-free facility and given food and water ad libitum. Animals received tamoxifen subcutaneously at 8-9 weeks of age and underwent facial nerve transection two weeks later.

This study did not involve wild animals.

This study did not involve field-collected animals.

Optogenetics experiment: The animals used were male CD® (Sprague Dawley) IGS Rats (Charles River), injected at the age of 8 weeks and sacrificed at 11-13 weeks of age.

The antibody has been tested by ELISA and/or solid-phase absorbed to ensure minimal cross-reaction with bovine, chicken, goat, guinea pig, Syrian hamster, horse, human, rabbit, rat, and sheep serum proteins, but may cross-react with immunoglobulins from other species. [Sigma-Aldrich]

Microglia experiment: As provided on manufacturer's website (see also [23] Tay, T. L. et al. Neuroscience 20, 793–803 (2017))



Animals and other organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research



Laboratory animals



Wild animals



Field-collected samples



|  |
| --- |
| nature research | reporting summary |



|  |
| --- |
| *April 2018* |

4