

Segmentierungsnetzwerke

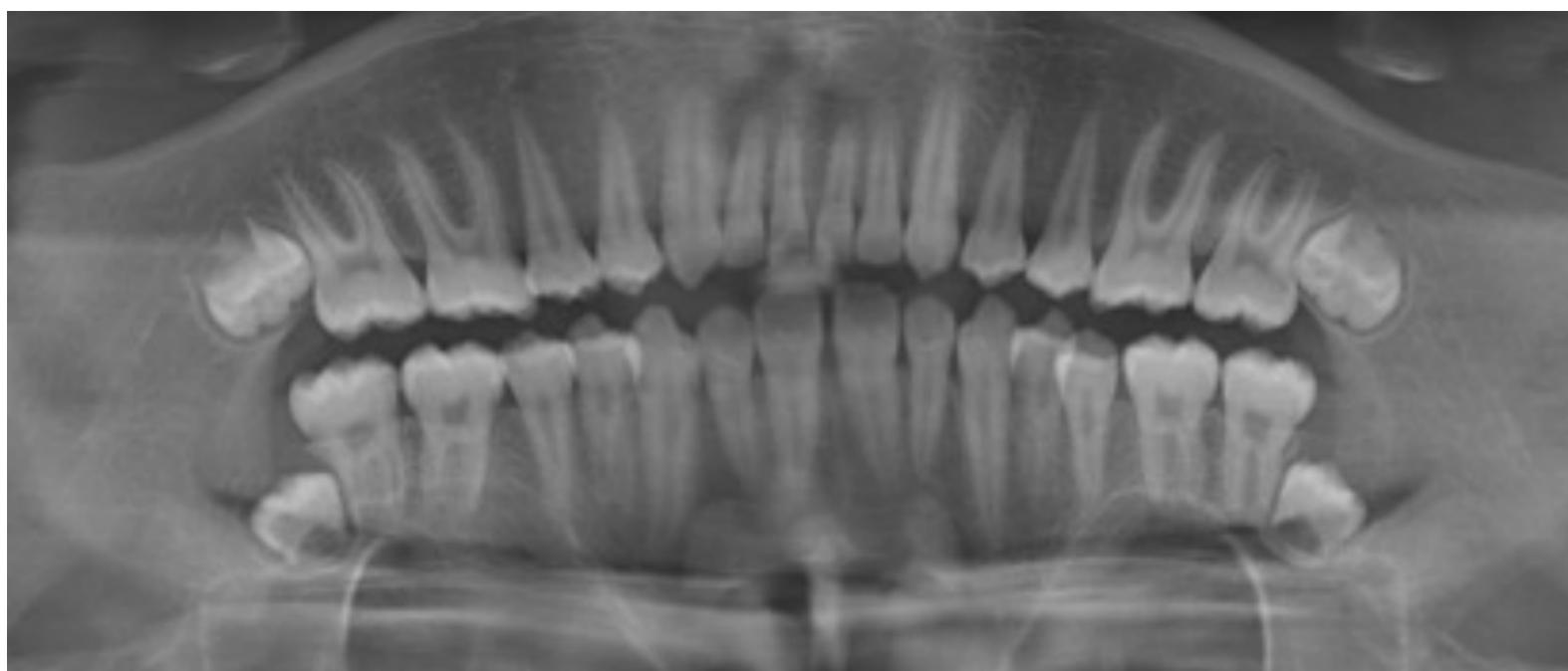
1. U-Net
2. DeepLab



Bildklassifikation

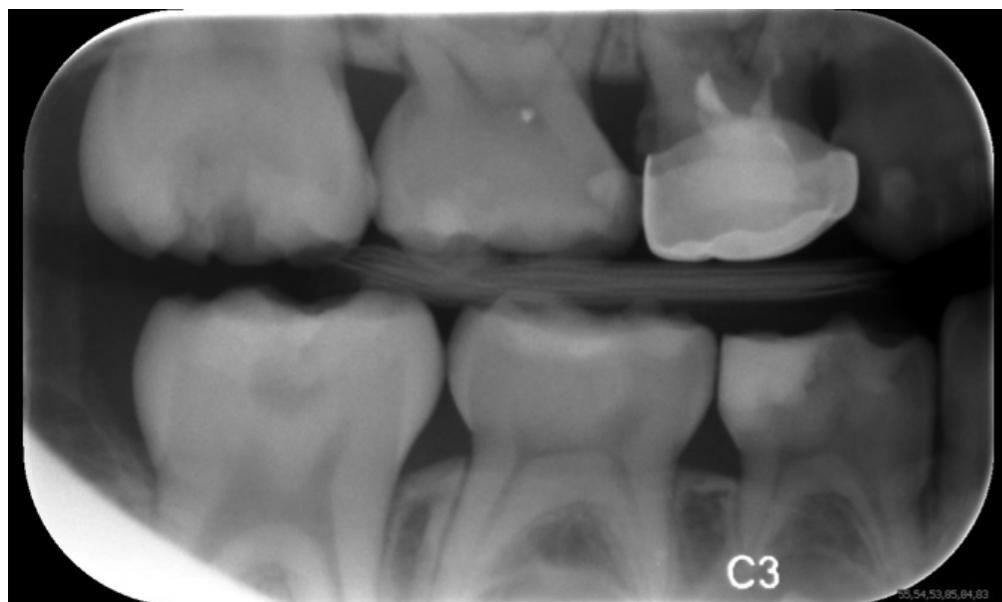
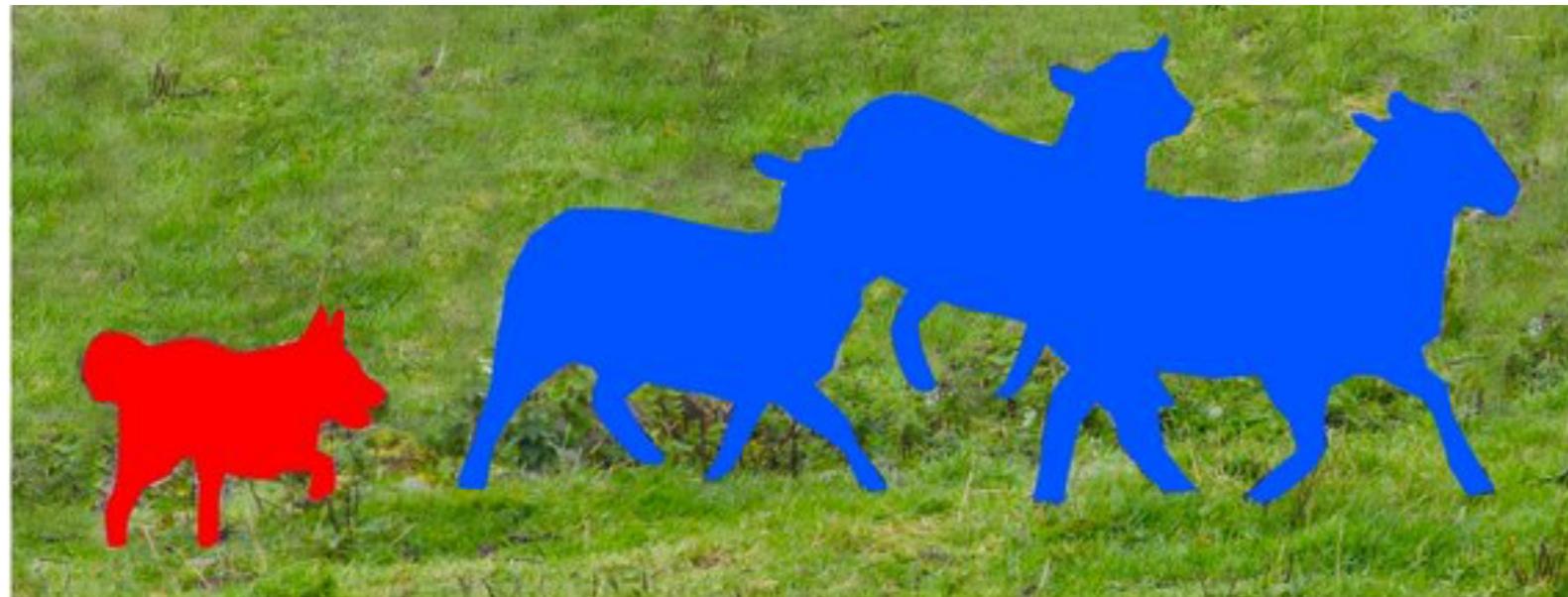


© ai-pool.com



© Jader et al., 2018

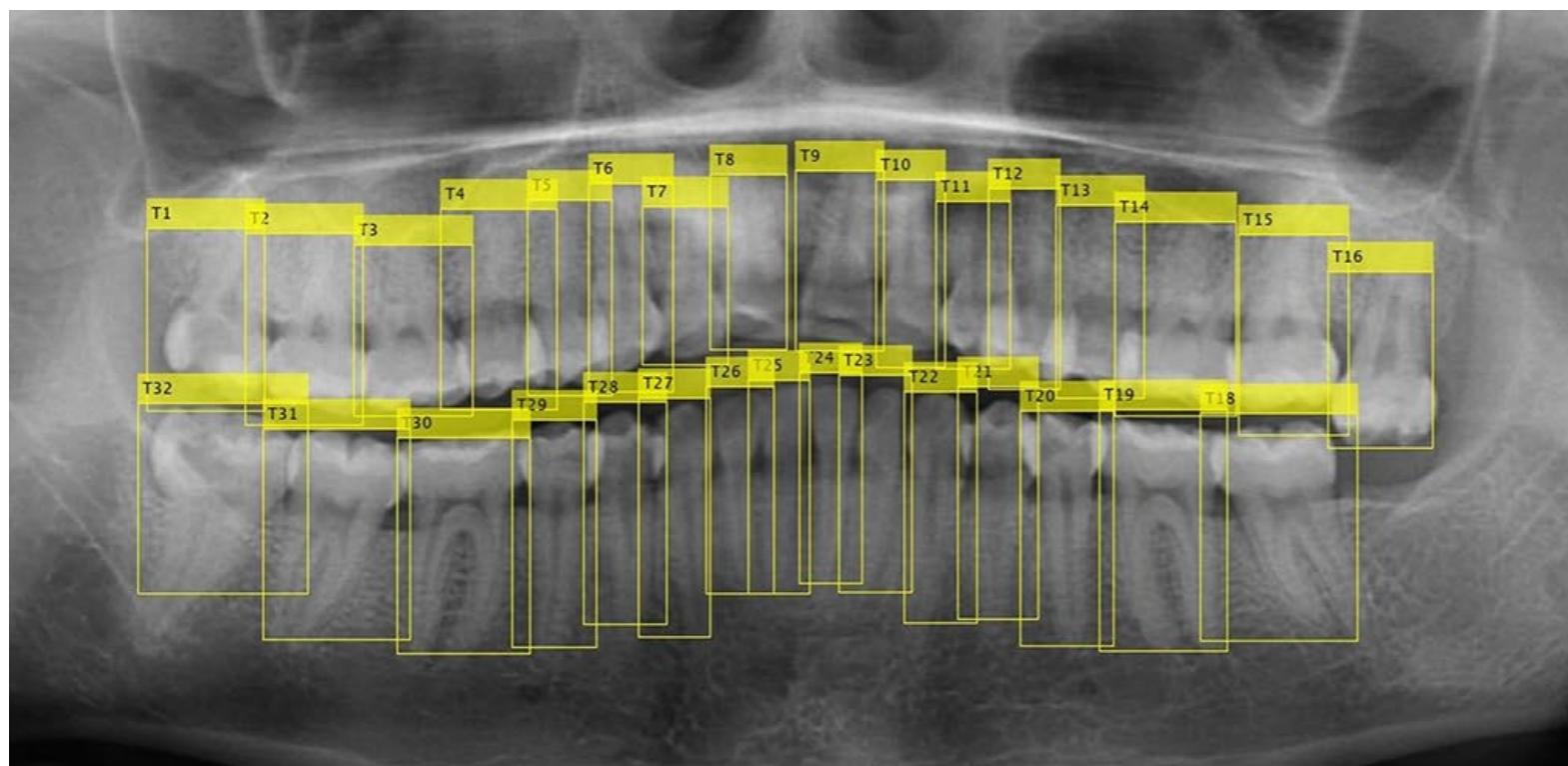
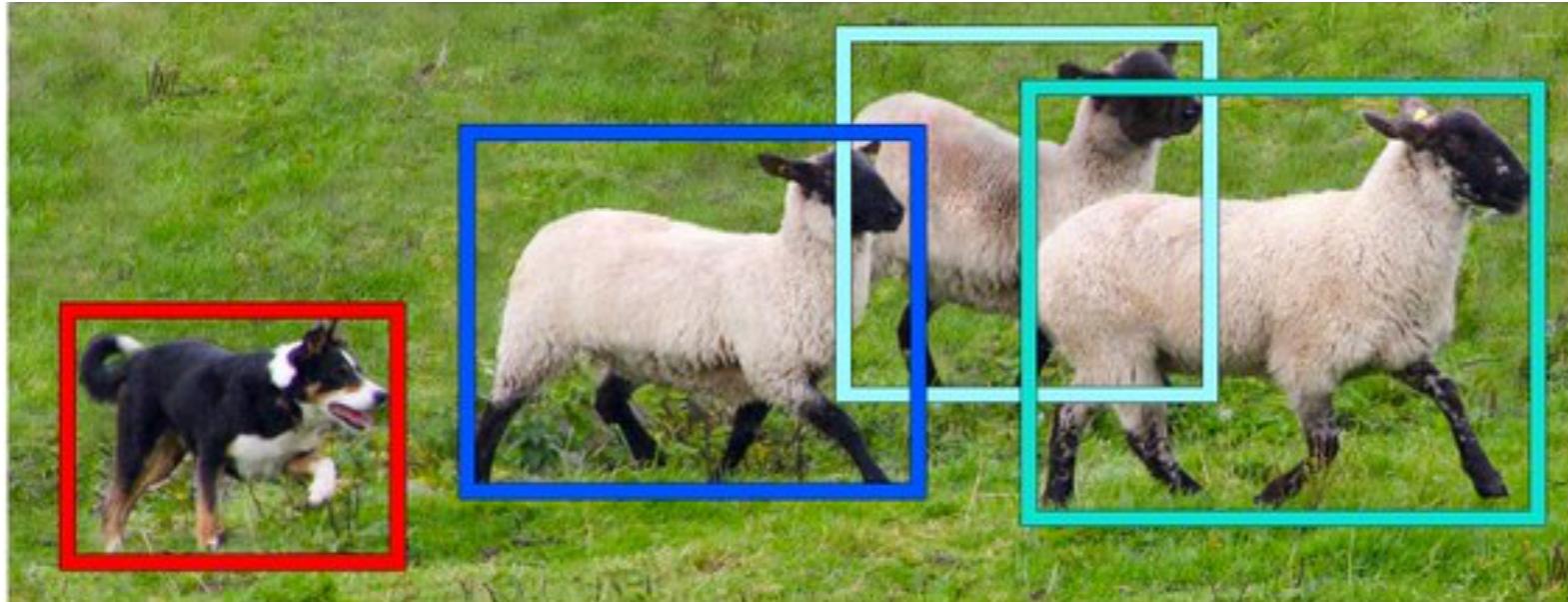
Semantische Segmentierung



© Ronneberger et al., 2015

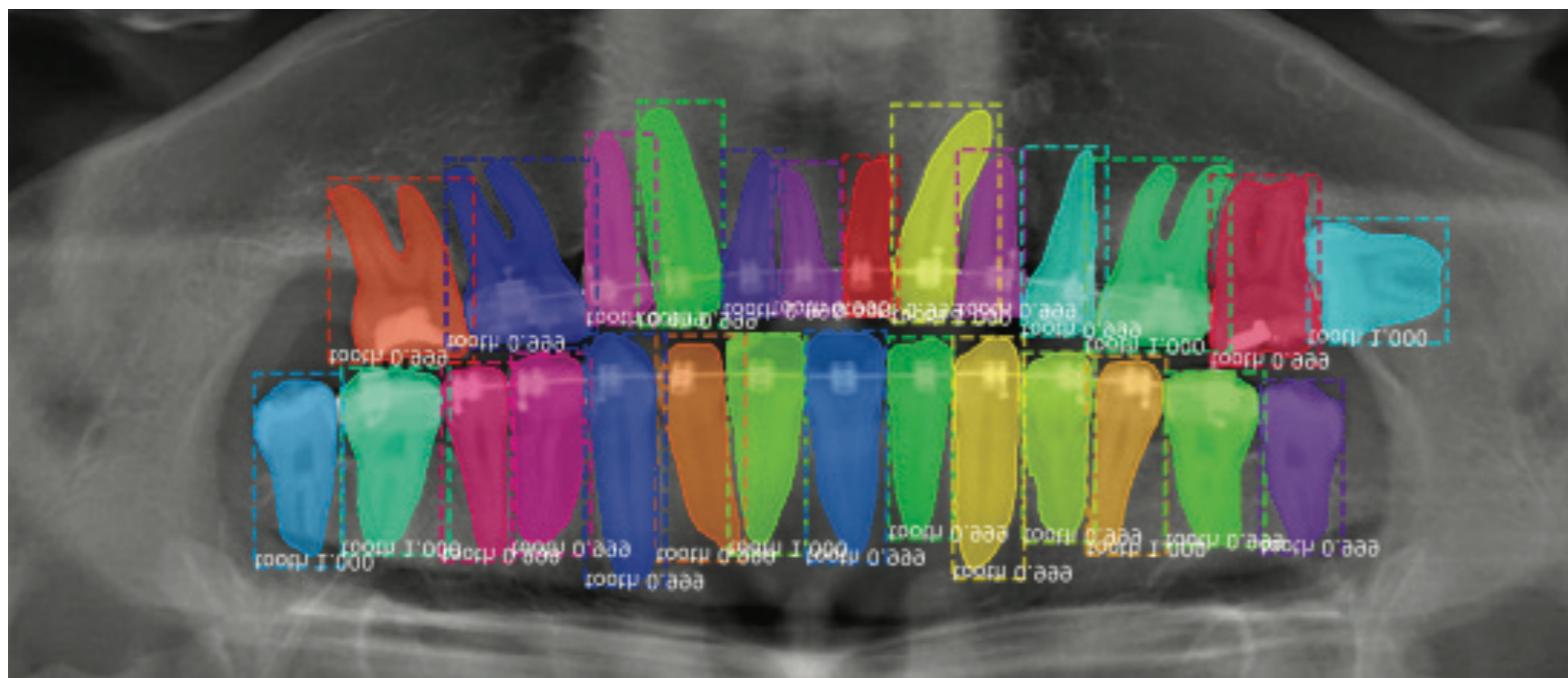
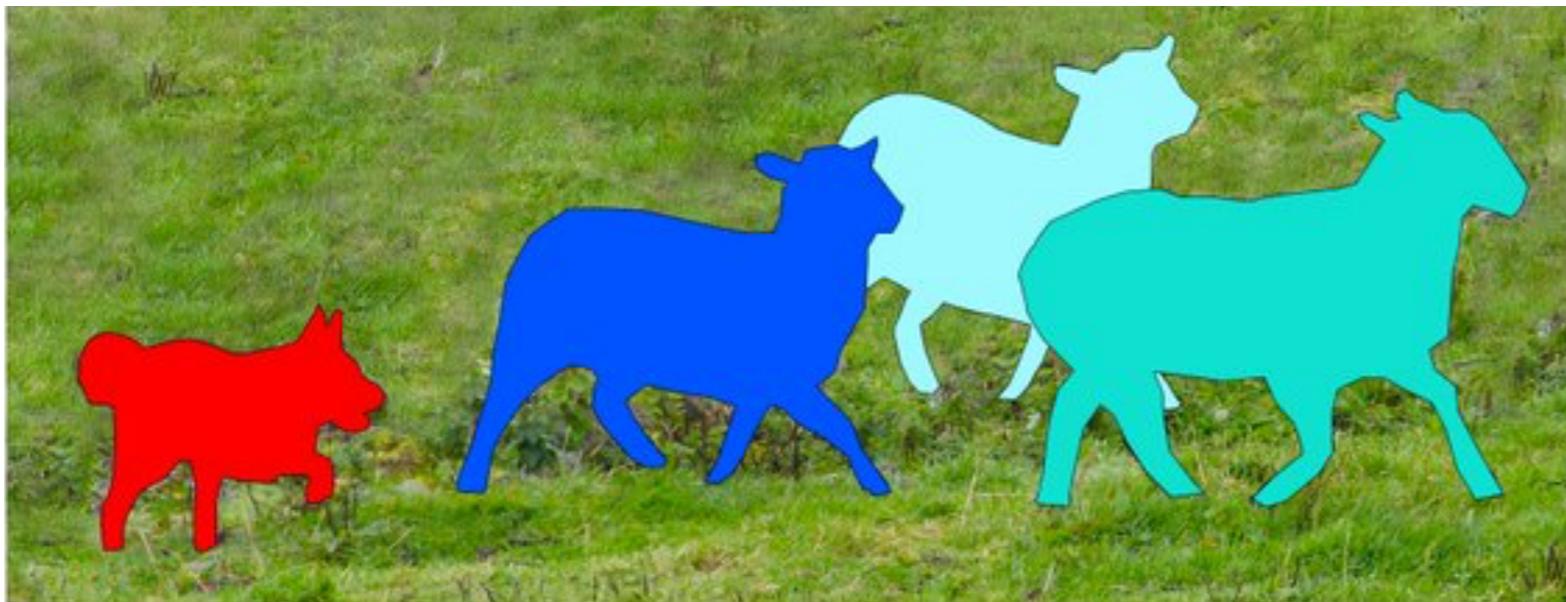
Karies (blau)
Schmelz (grün)
Zahnbein (gelb)
Mark (rot)
Krone (rosa)

Objektdetektion



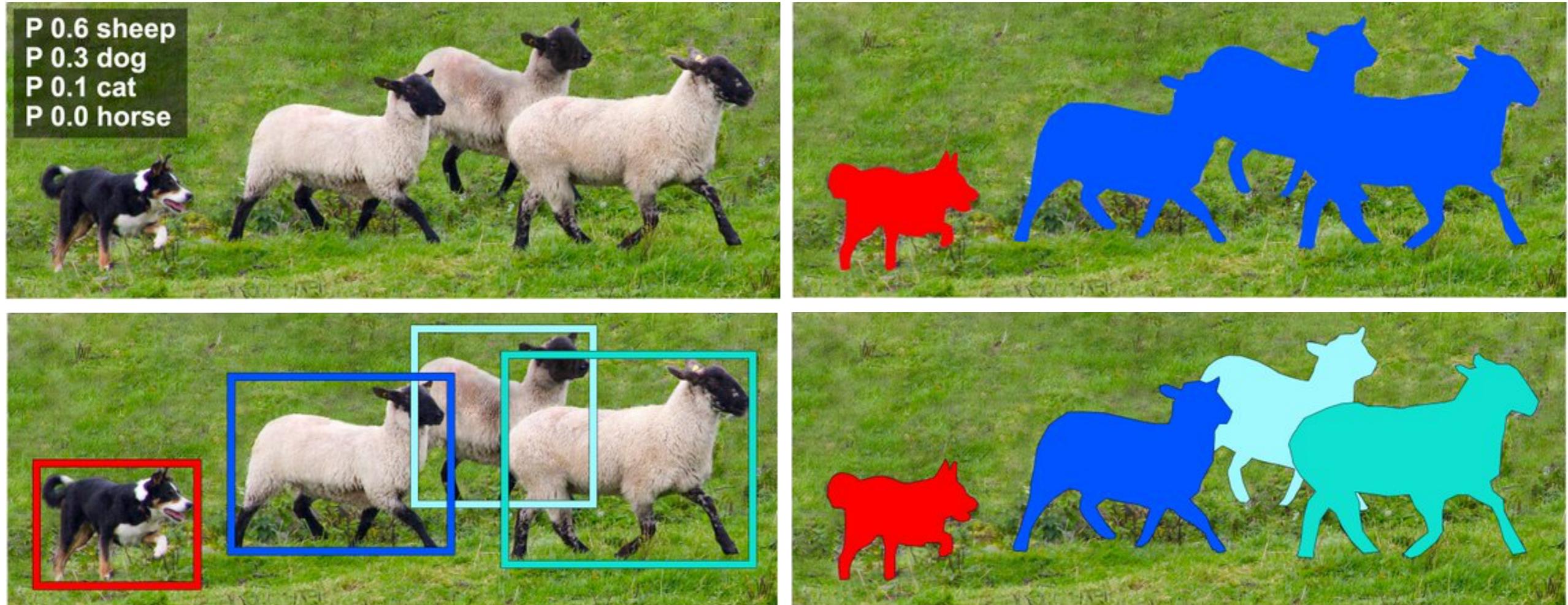
© Mahdi 2020

Instanzsegmentierung



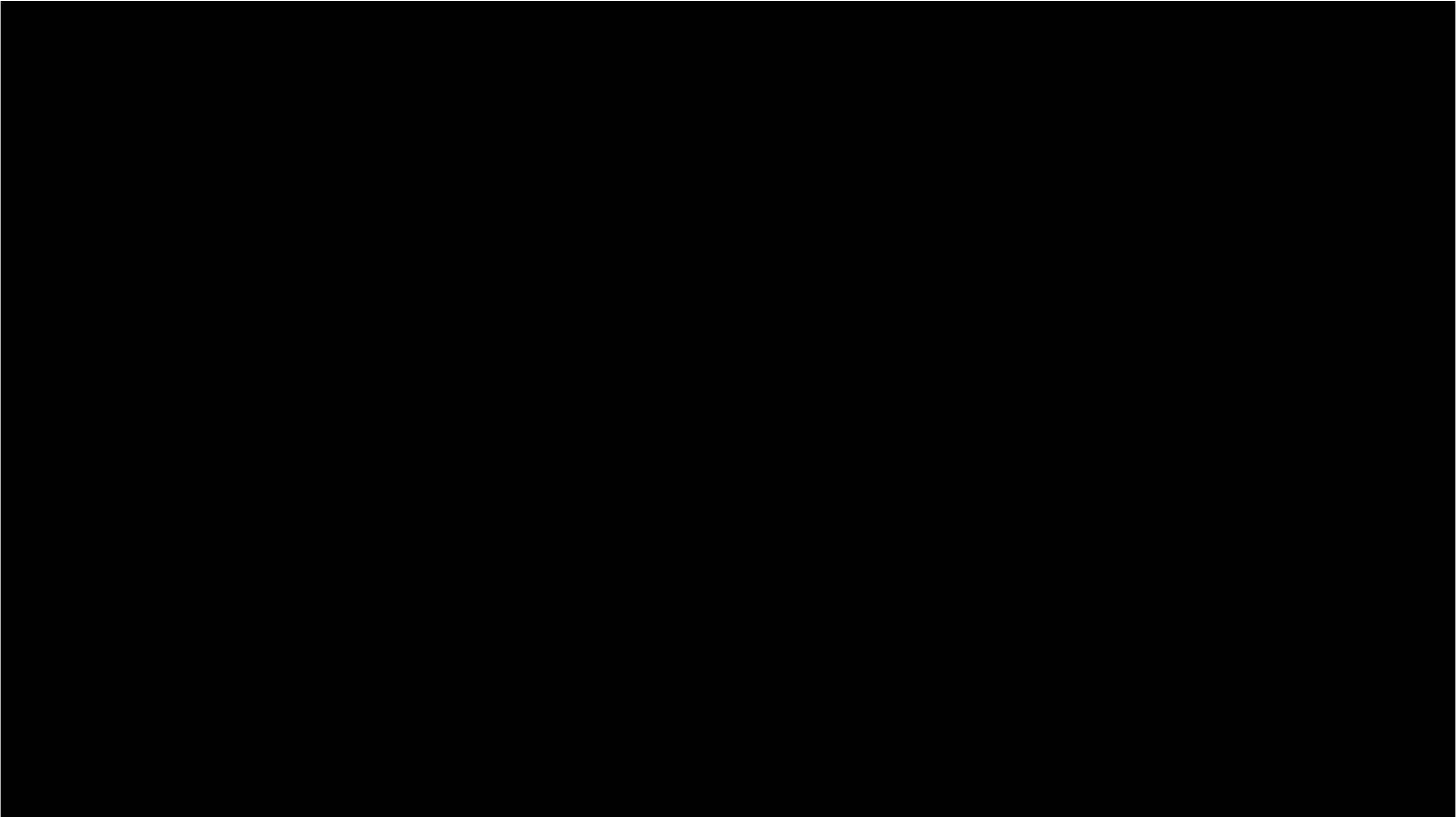
© Jader et al., 2018

Auf einen Blick



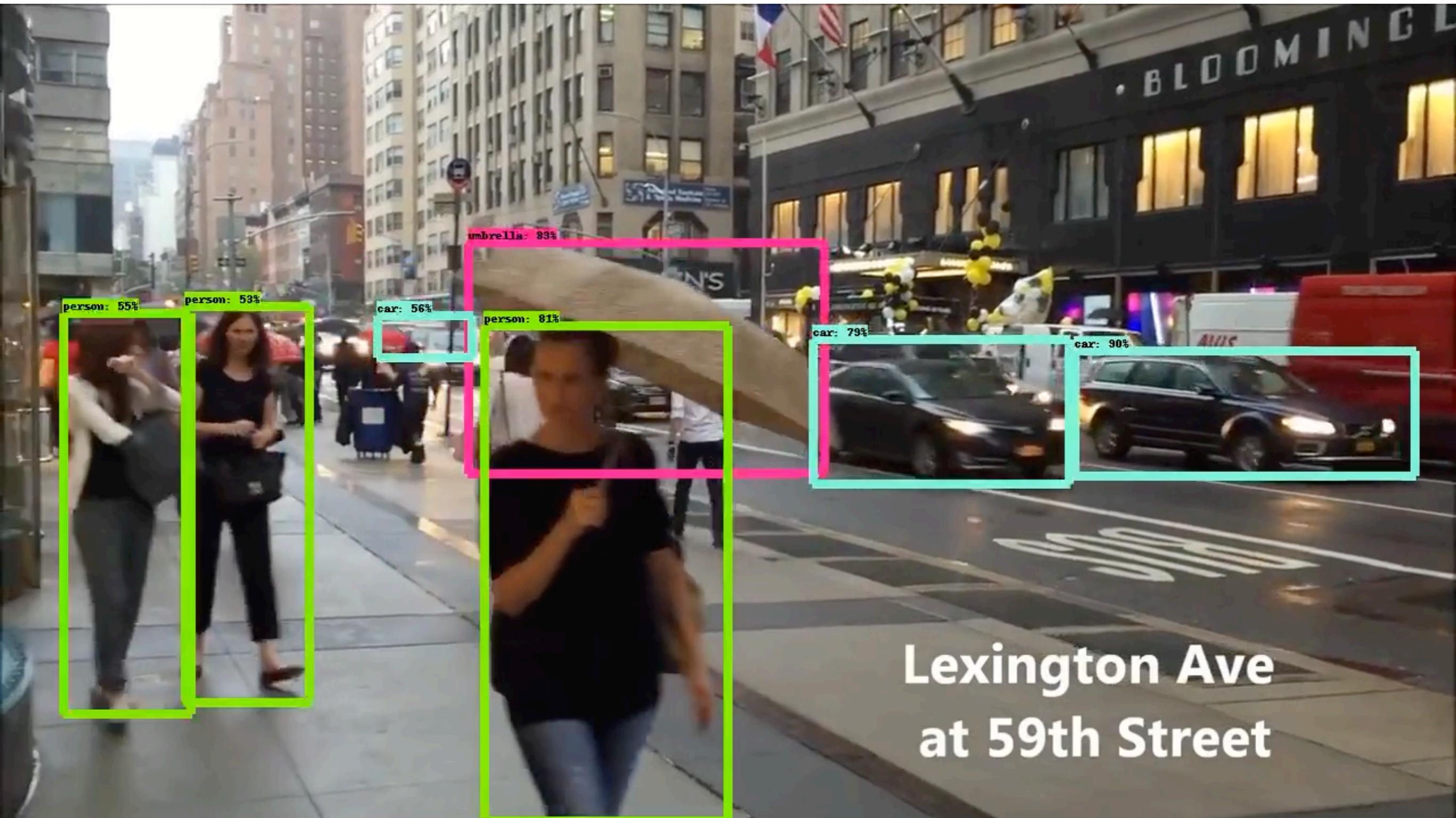
© ai-pool.com

Beispiel Bildklassifikation



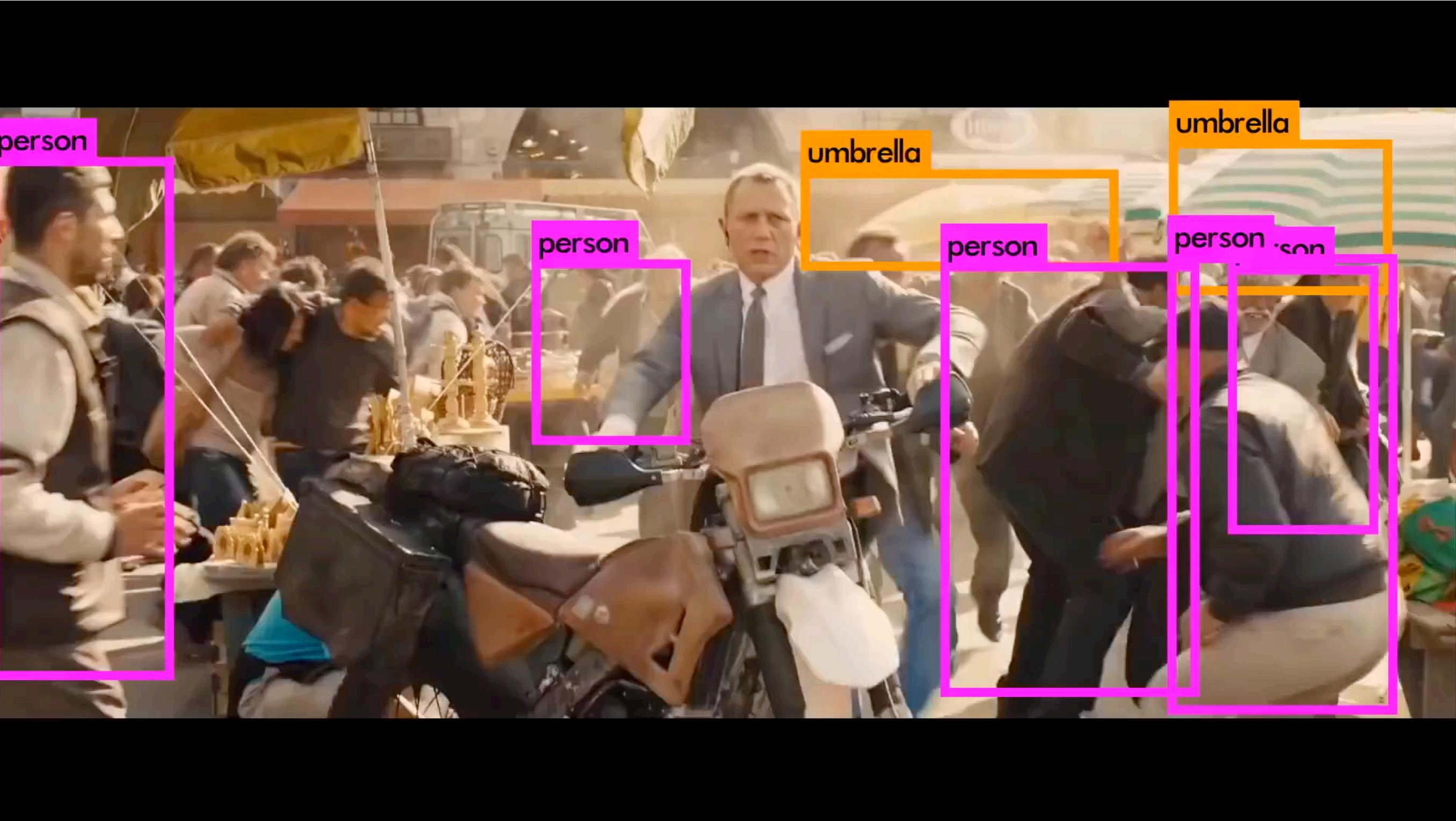
© Misawa et al., 2015

Beispiel Objektdetektion



©Diego Cavalca, https://www.youtube.com/watch?v=_zZe27JYi8Y&t=0s

Beispiel Objektdetektion



©pjreddie.com, <https://www.youtube.com/watch?v=VOC3huqHrss&t=0s>

Beispiel Semantische Segmentierung

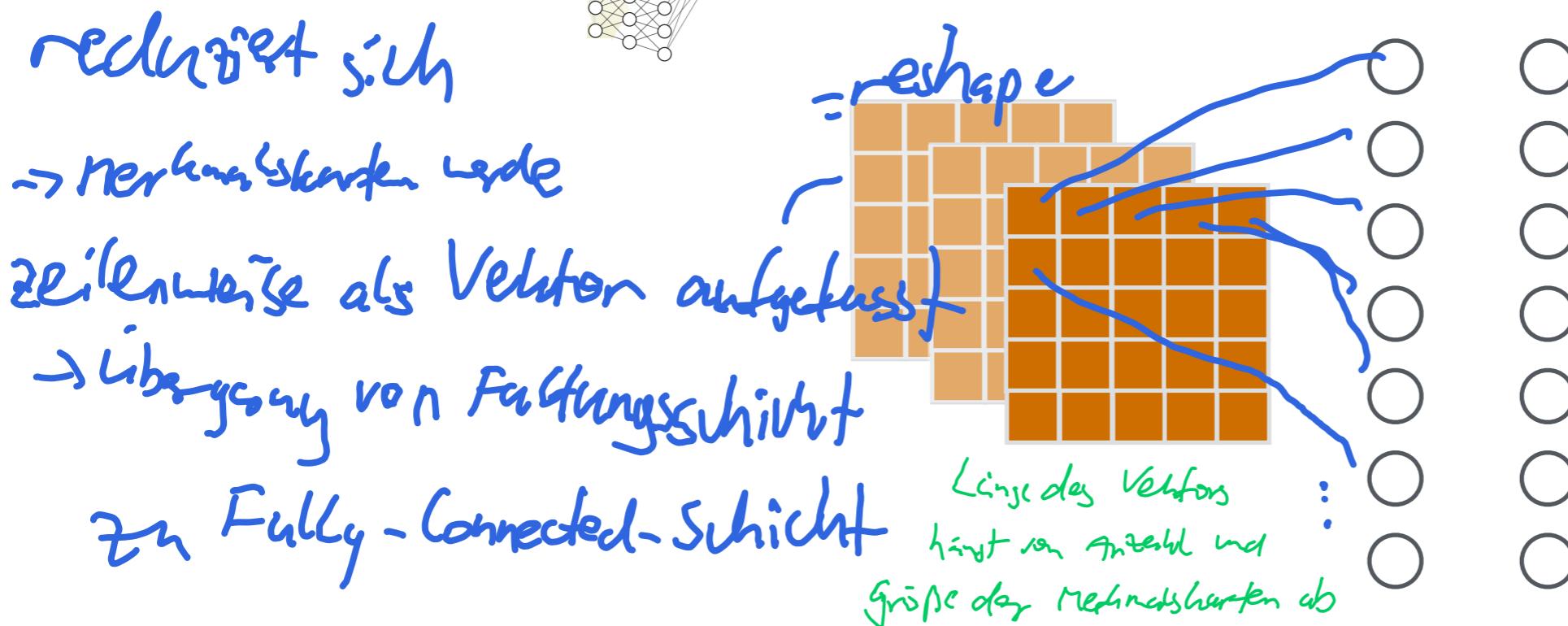
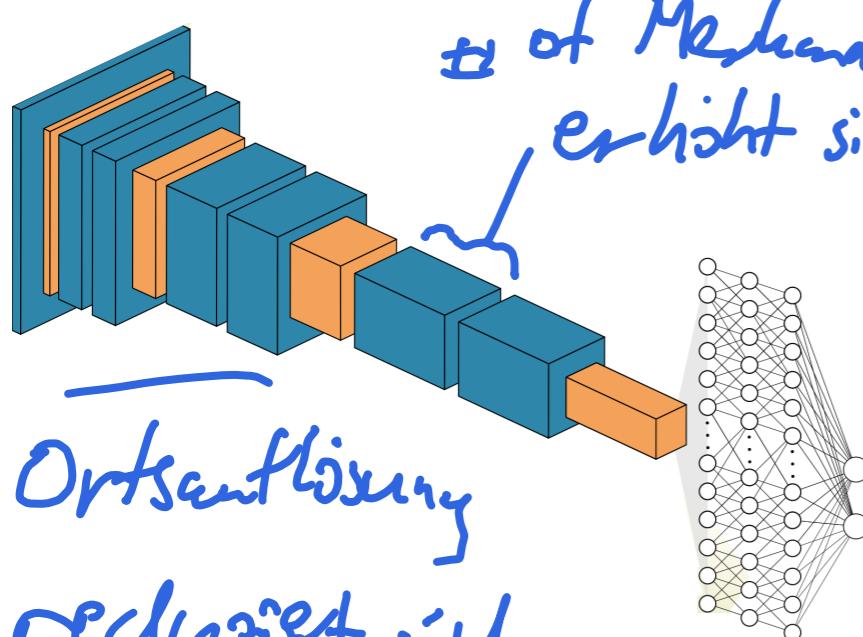


©Zhao et al, <https://www.youtube.com/watch?v=qWI9idsCuLQ&t=0s>

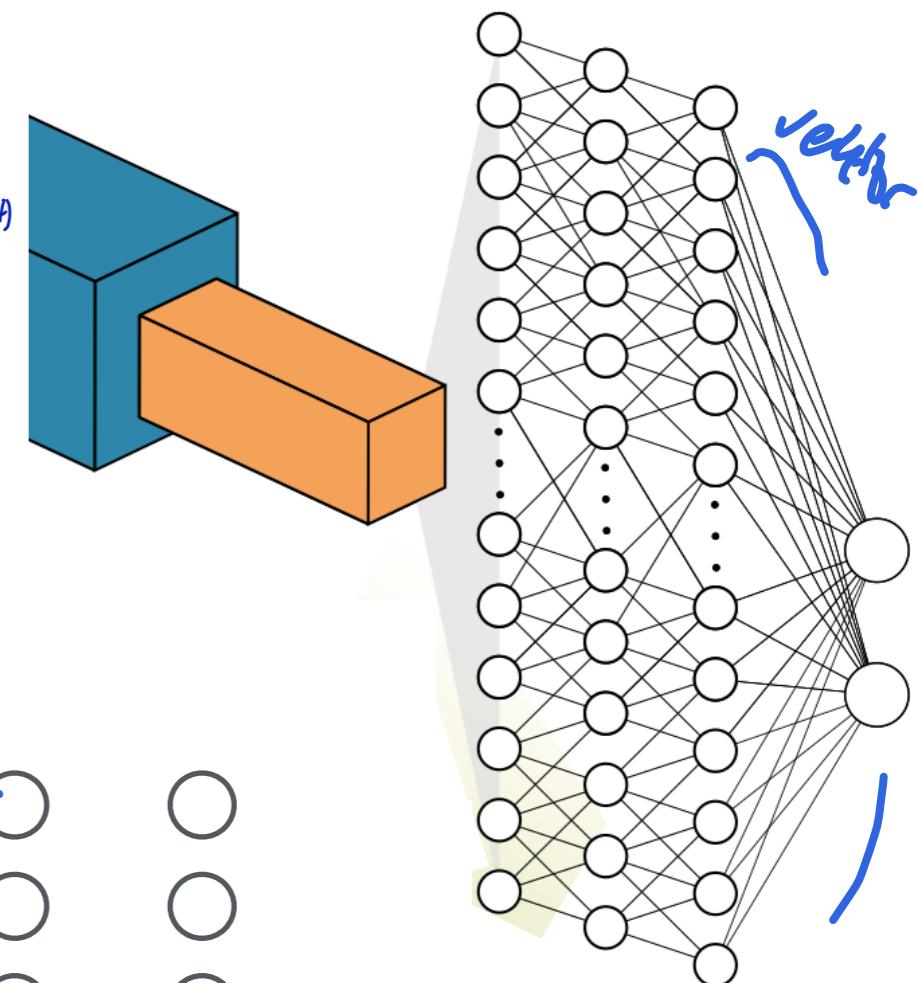
Klassifikation

Convolutional Neural Networks

mit Fully Connected Schichten als Klassifikator



Fully Connected - Teil



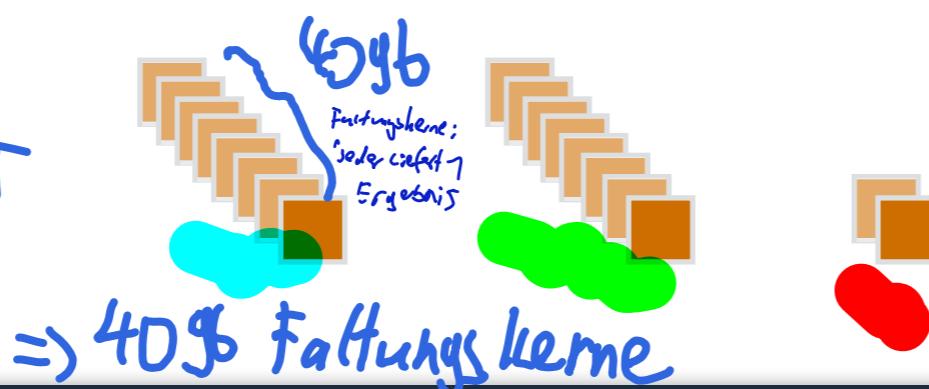
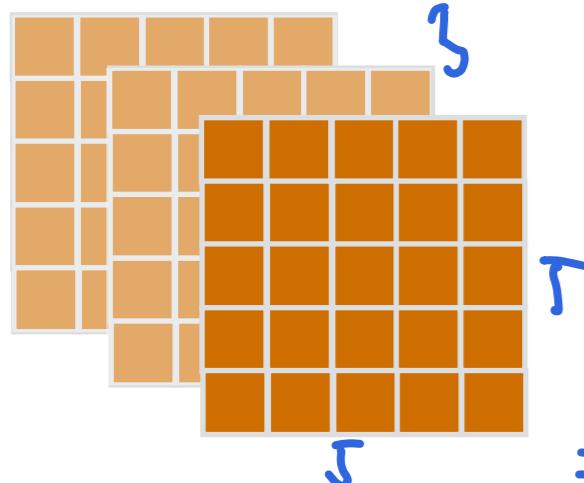
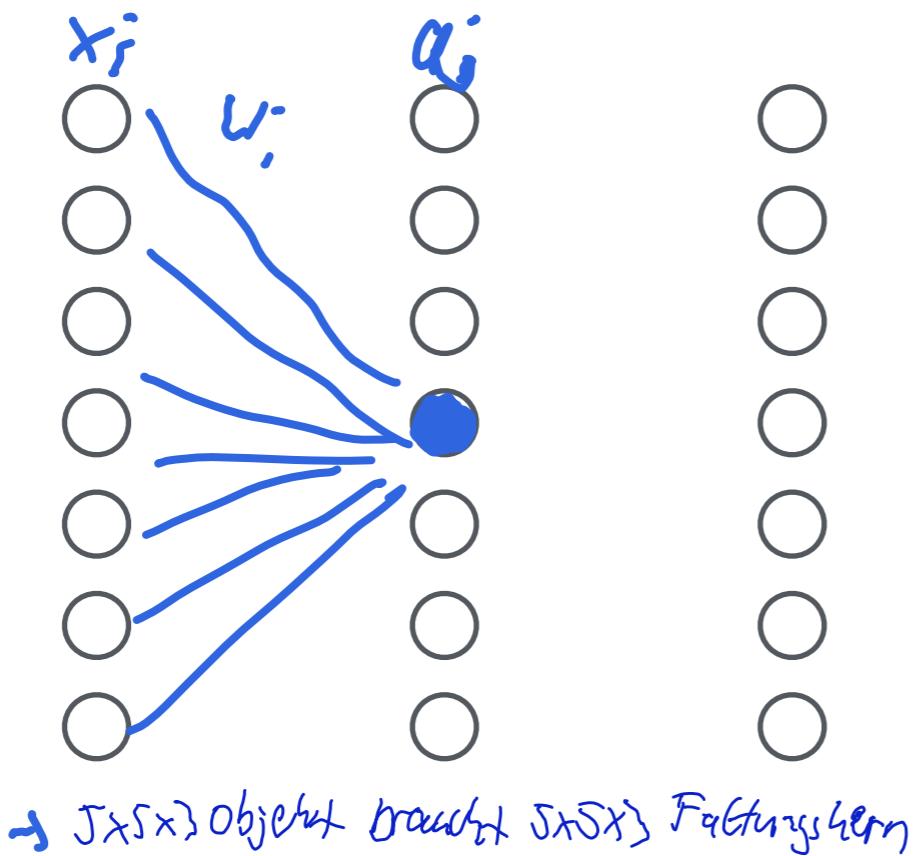
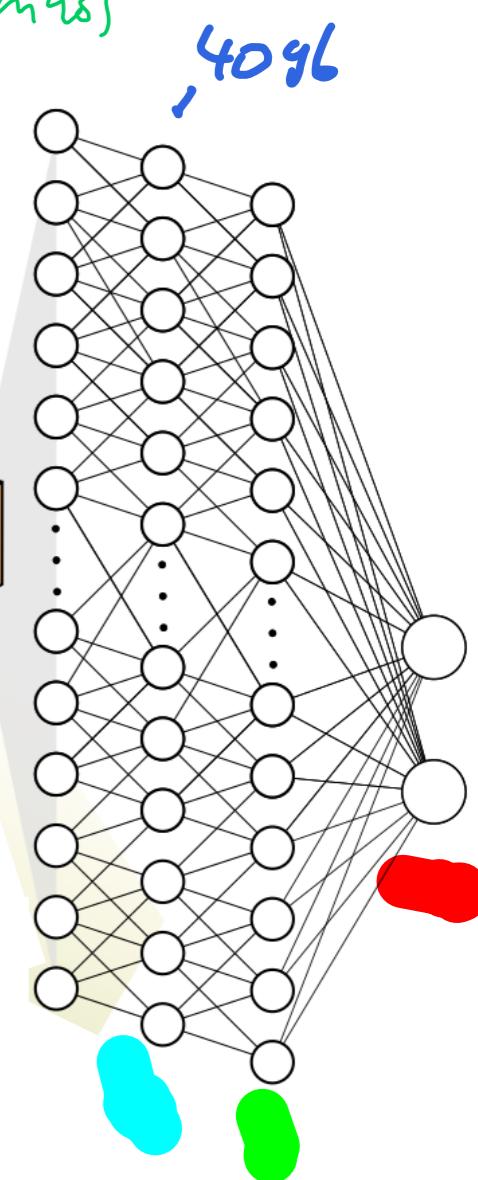
2-Klassen-Problem

Klassifikation

Fully Convolutional Neural Networks

man spart sich reshape, muss
aber Faltungskerne lassen

Fully Connected Schicht als Faltung interpretieren (5×5)



$$q_j = \sum w_{ij} x_i$$

↪ Aktivierungsfkt. zur
Vereinfachung vergessen

Klassifikation

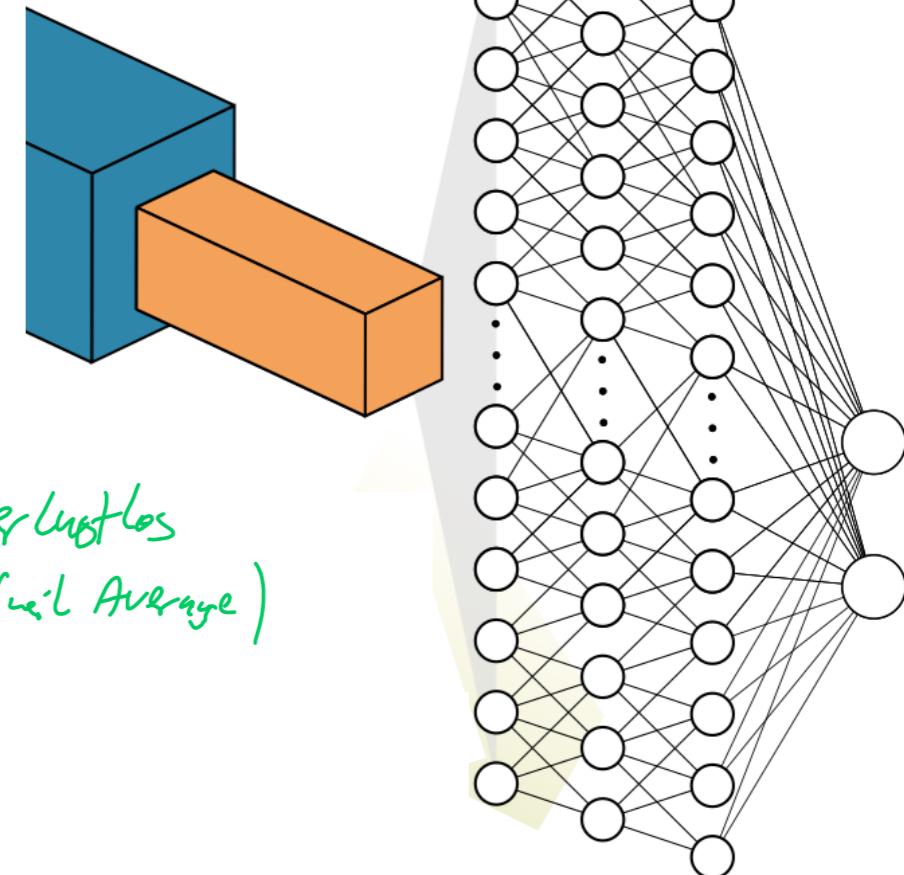
Fully Convolutional Neural Networks

Fully Connected Schicht als Faltung interpretieren

... von der Bildgröße unabhängig

↪ Global Average Pooling

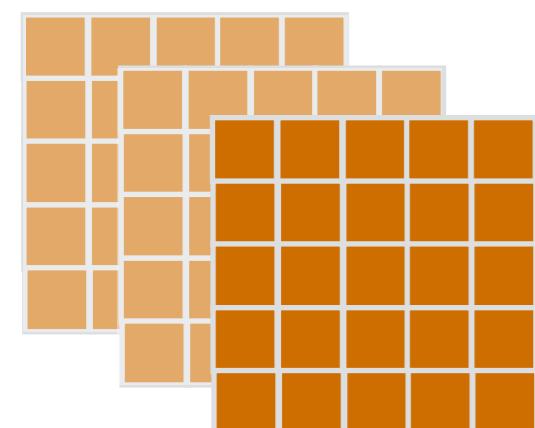
⇒ Mittelwert über jeweils eine
merkmalshafte



Ist aber nicht verlustlos
→ GAP verliert Info (nur Average)

FC-Schicht als

Faltung simulieren



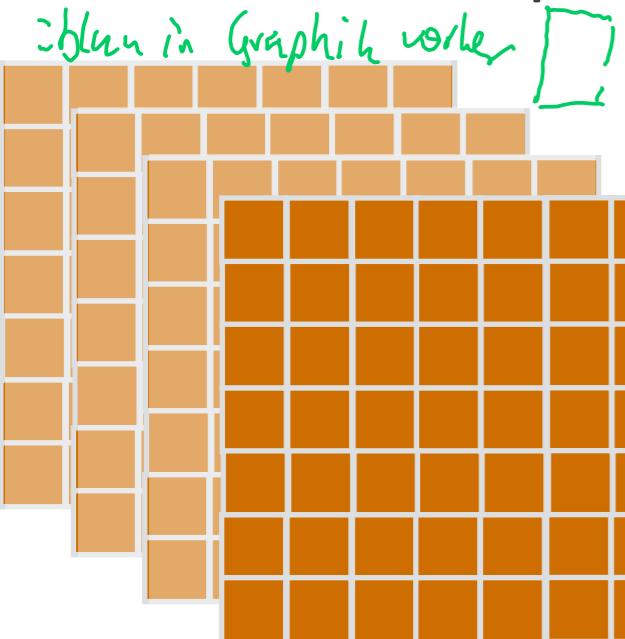
GAP



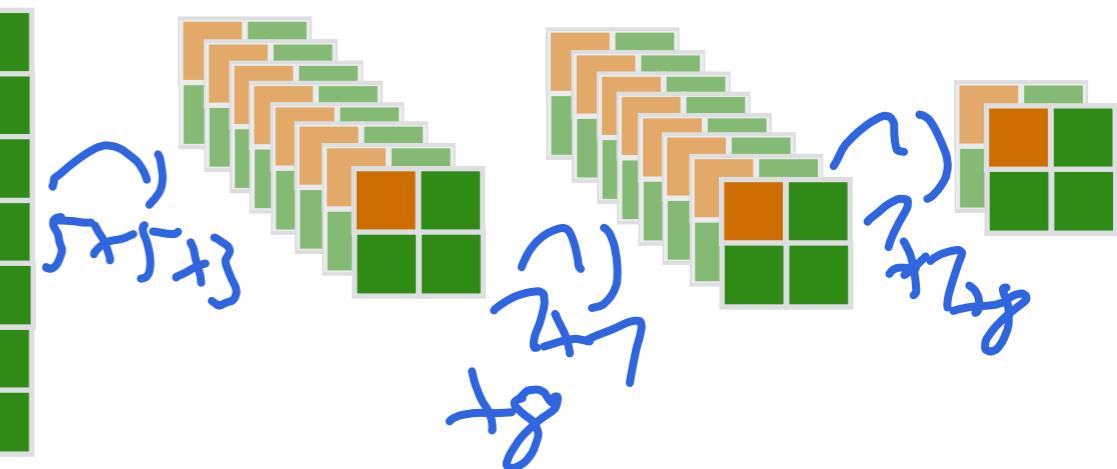
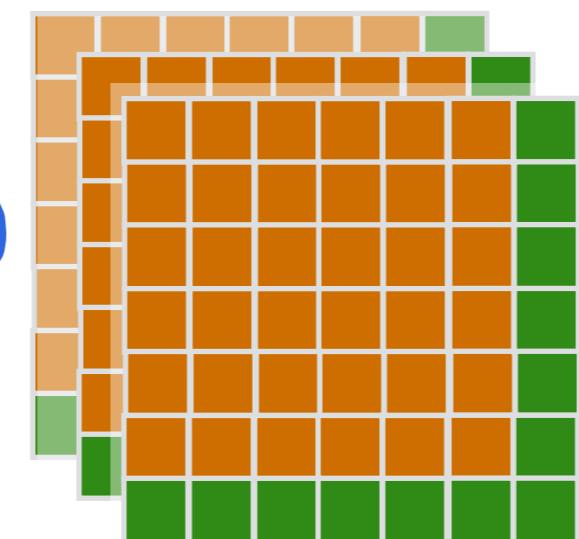
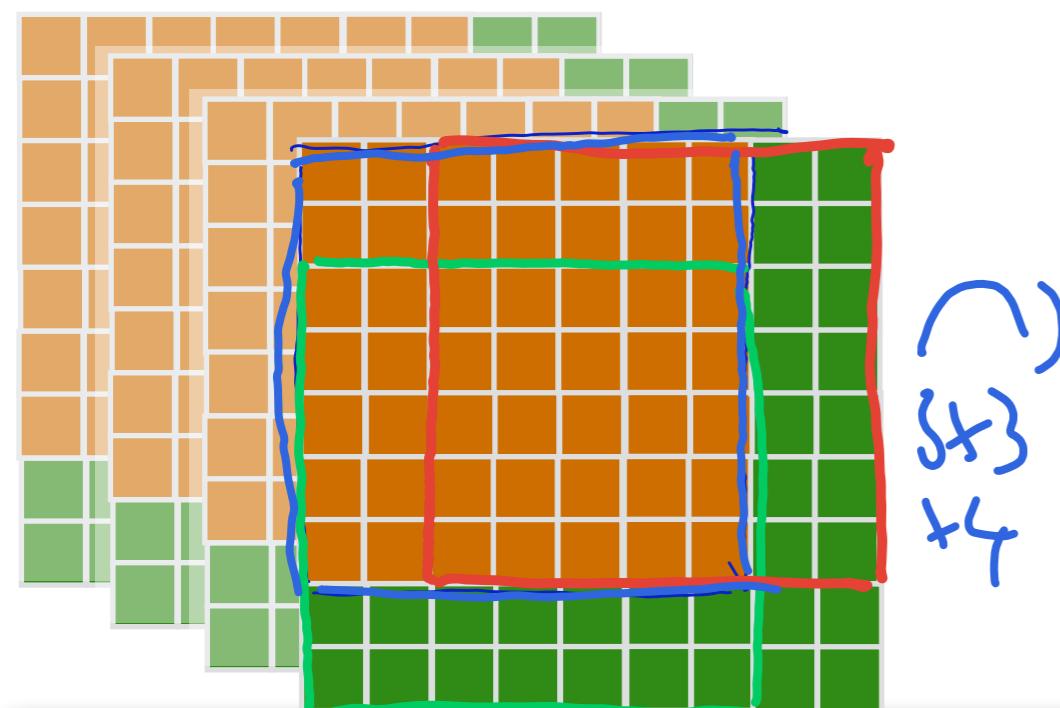
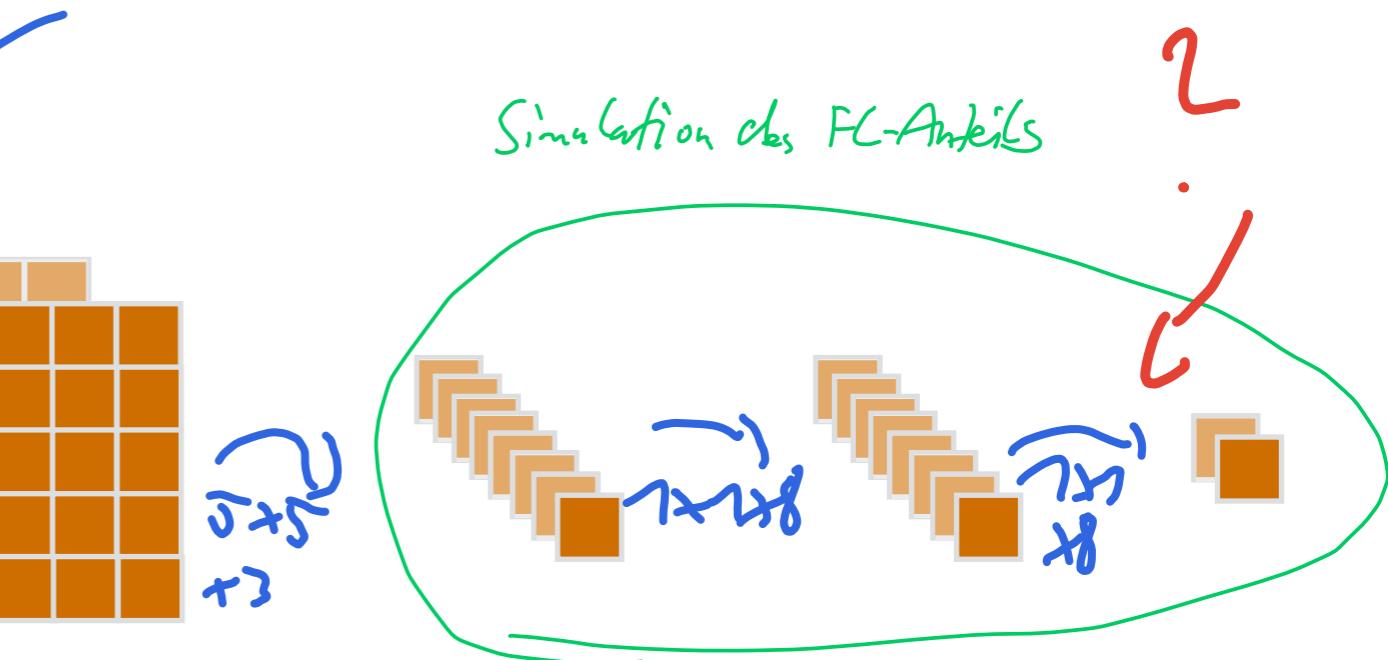
Sliding Window

... als Faltung

effiziente Implementierung



Ausschnitt eines Sliding Window



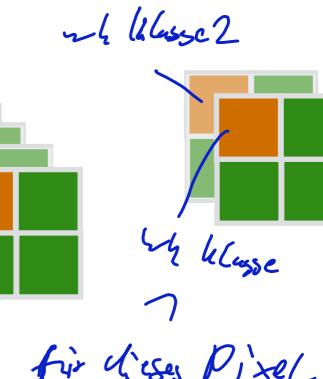
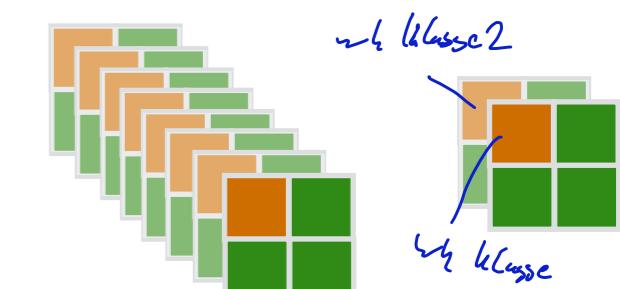
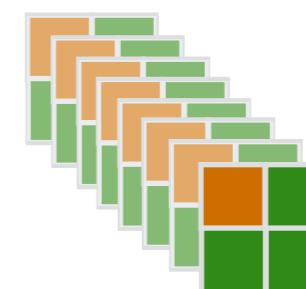
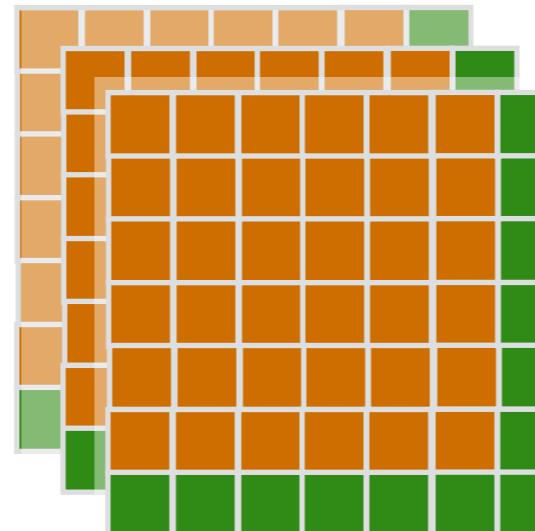
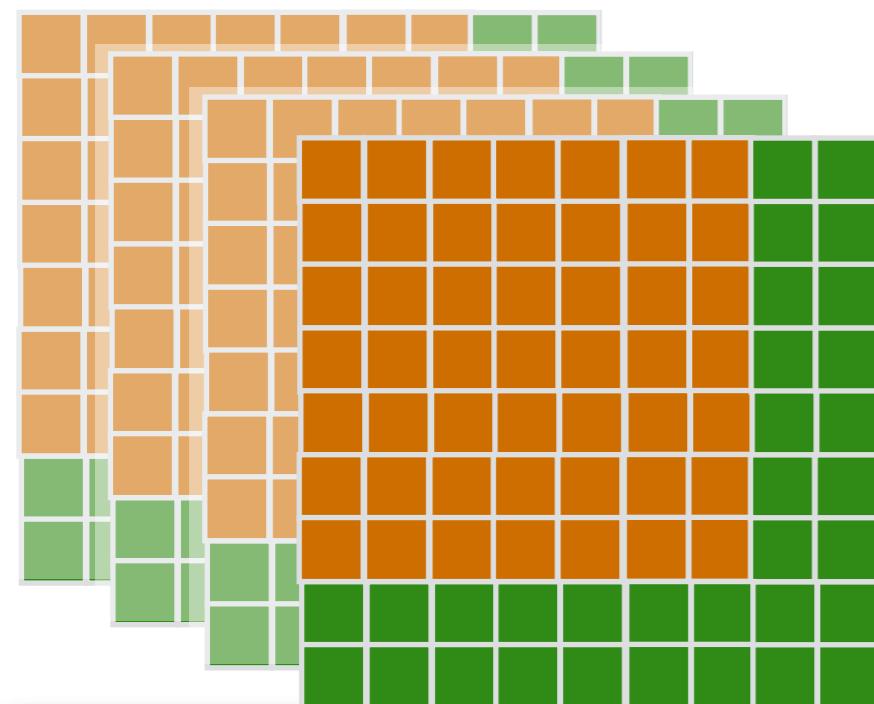
nach Sermanet et al., 2014

Sliding Window

... als Faltung

effiziente Implementierung

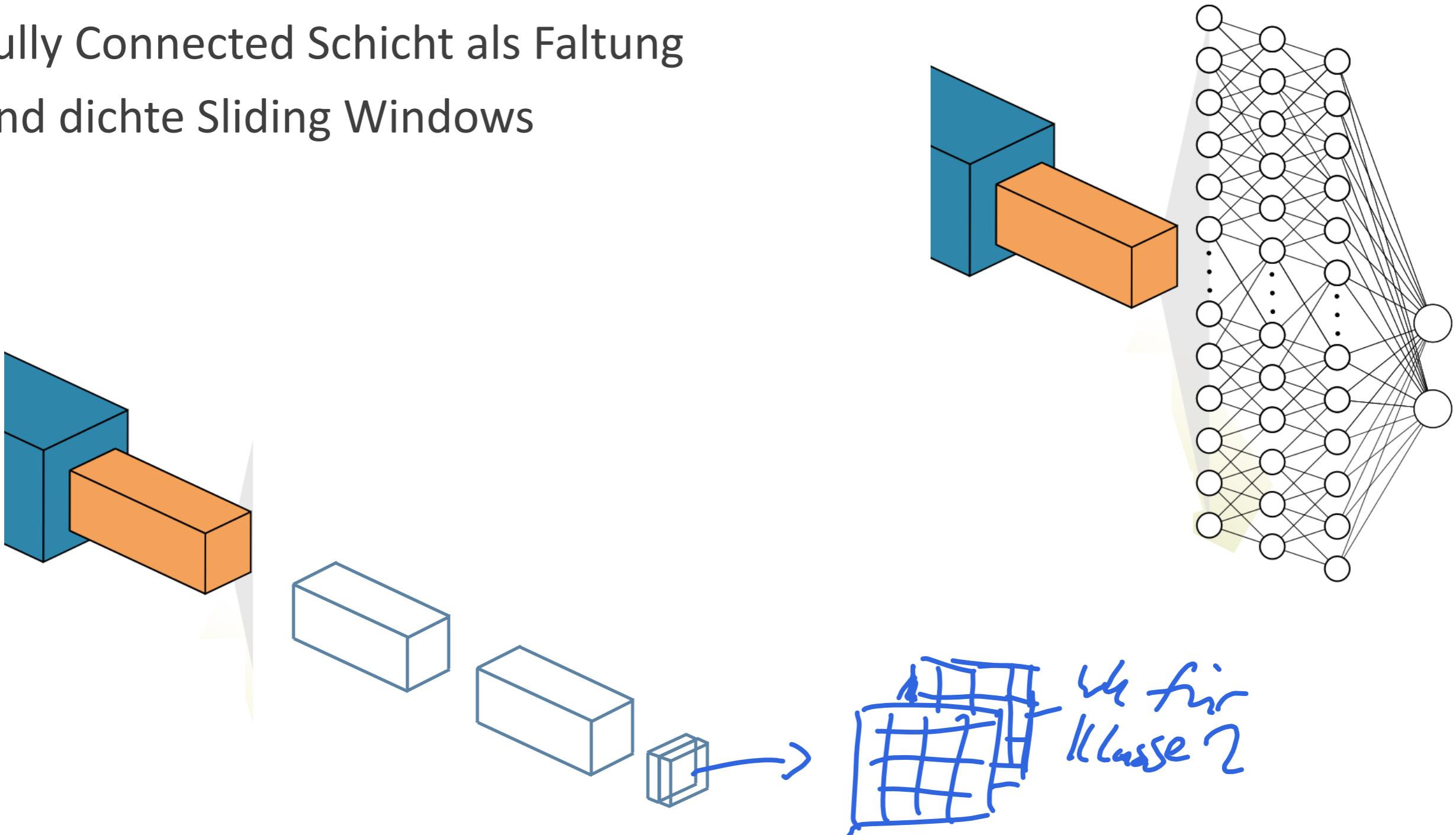
⇒ Segmentierungsresultat für jede Pixelposition



nach Sermanet et al., 2014

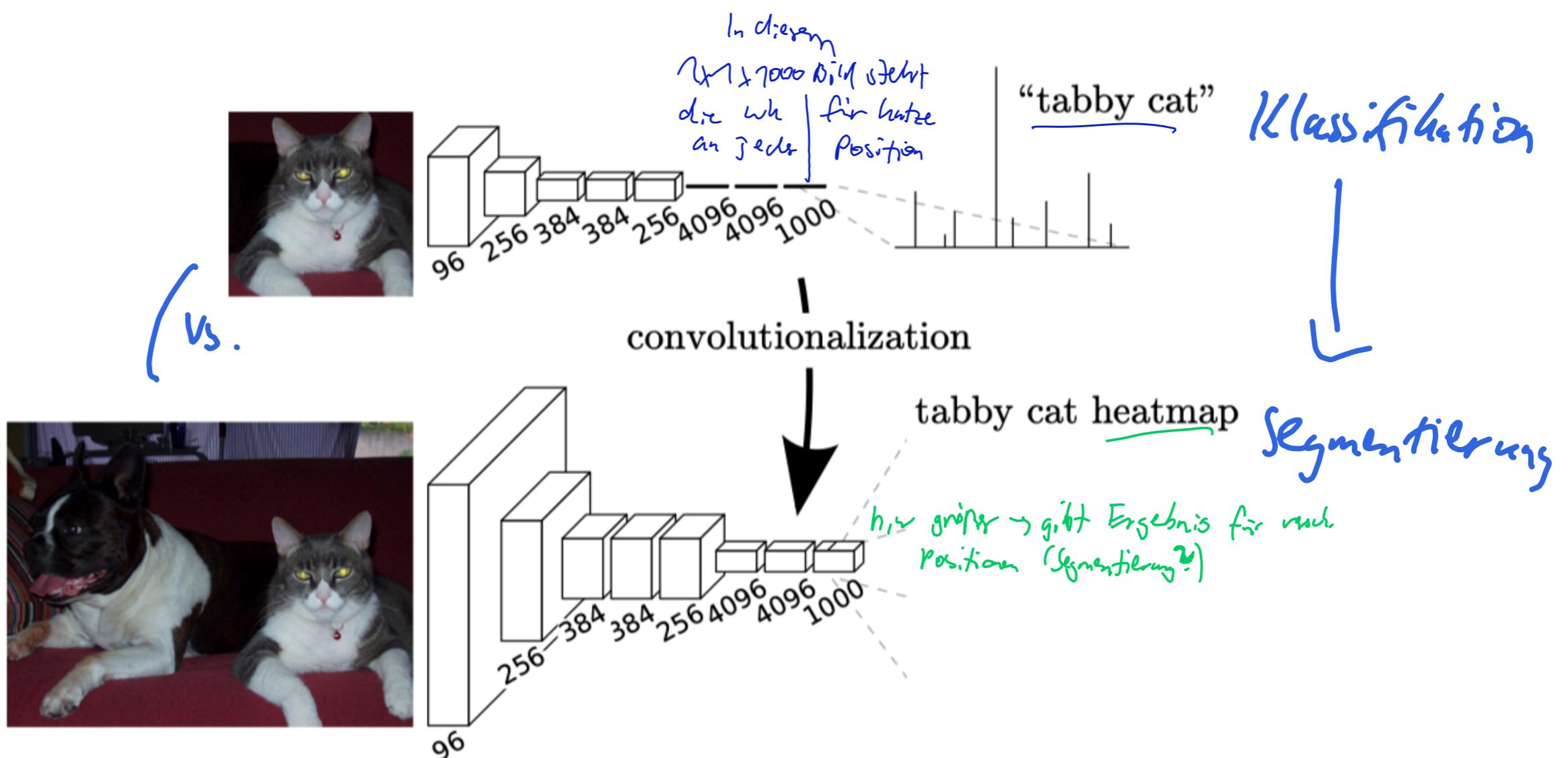
Semantische Segmentierung

Fully Connected Schicht als Faltung
und dichte Sliding Windows



Semantische Segmentierung

Fully Connected Schicht als Faltung
und dichte Sliding Windows



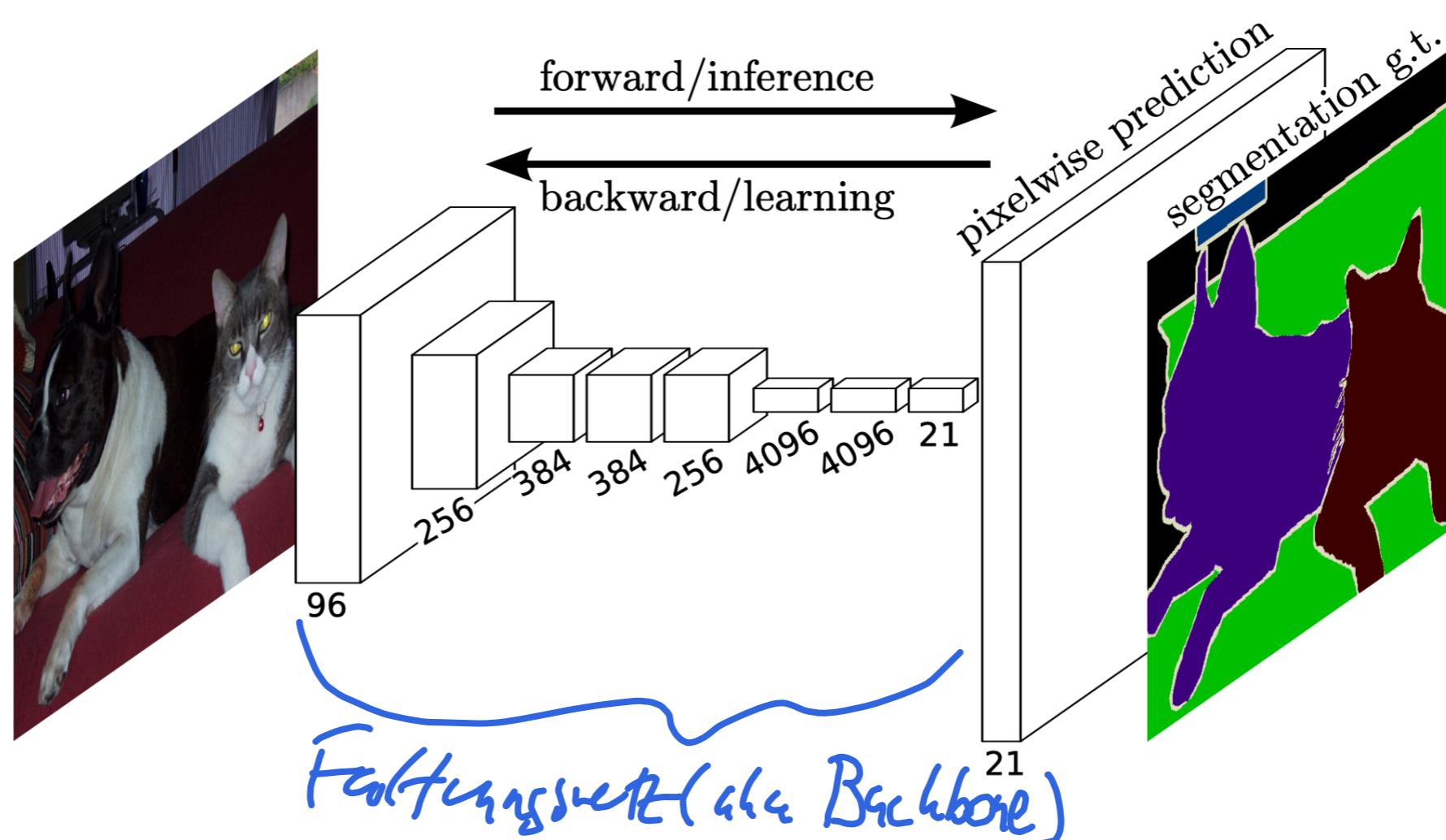
Semantische Segmentierung

Ziel

hochauflößtes Bild am Ende

Pixelgenaue Klassifikation

↳ UpSampling der LabelKarte

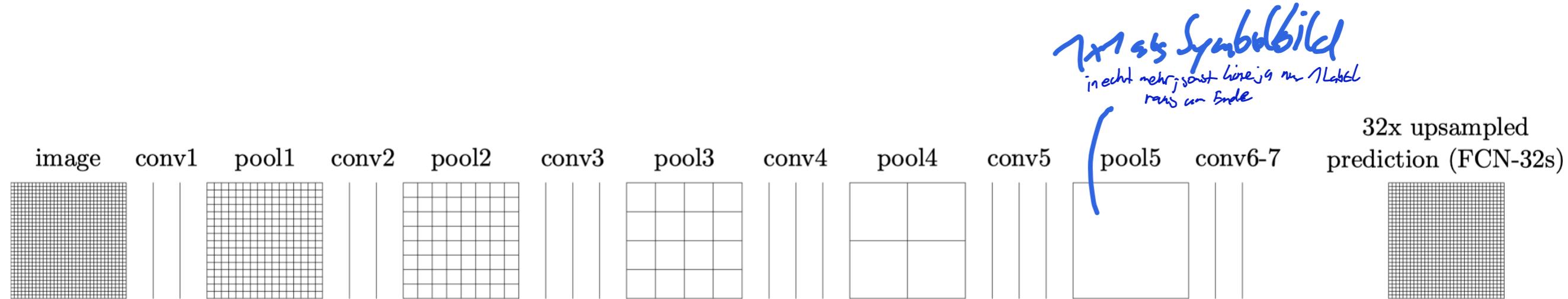


© Long et al., 2015

Semantische Segmentierung

Upsampling

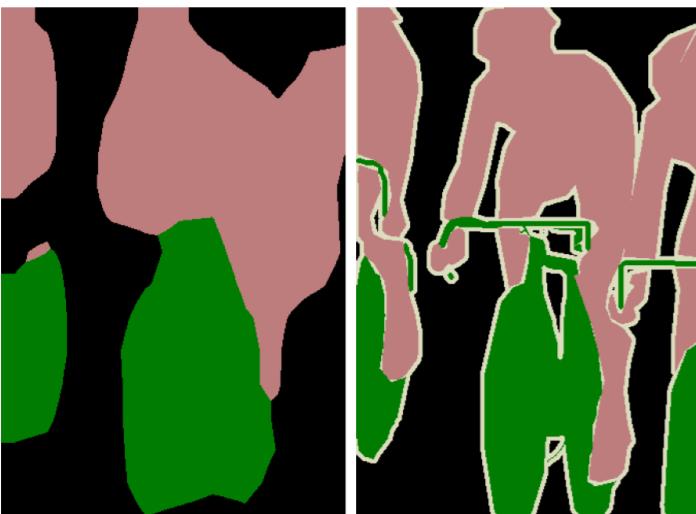
Du hir: FCN: Fully Convolutional Network



⇒ linear Interpolation → für Labelkarten
ungeeignet → Lösung: Nearest Neighbor Interpolation

Labelkarte
→ Label an jeder Pixelposition
der Labelkarte

FCN-32s Ground truth



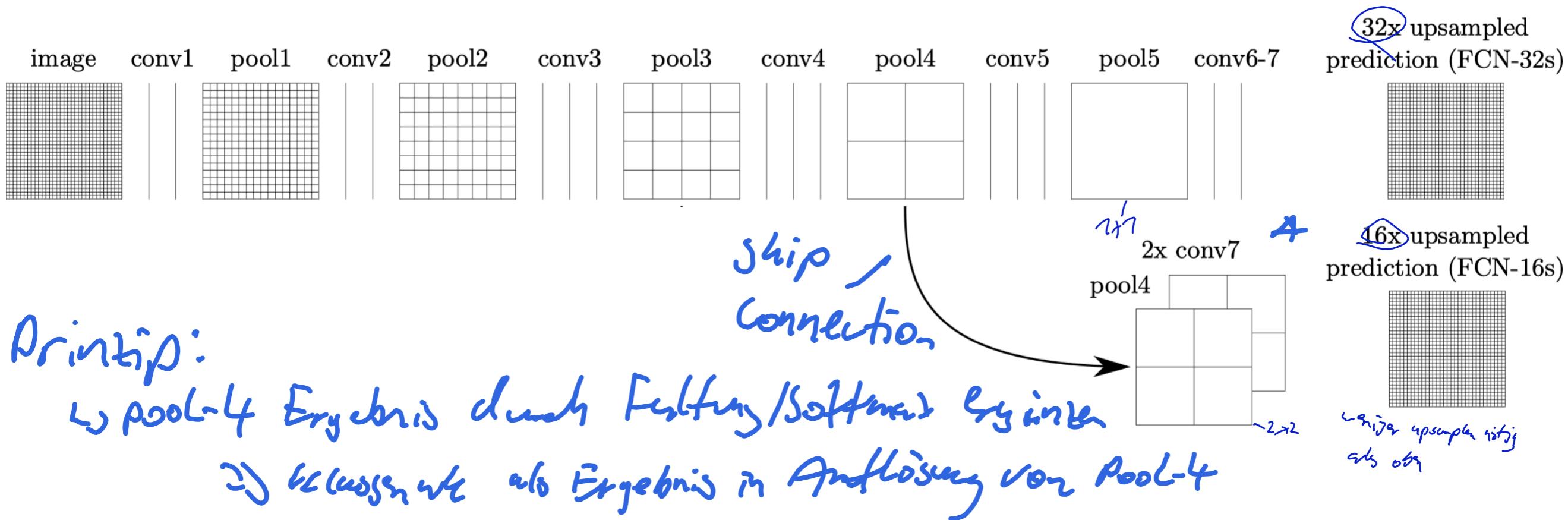
Interpolation z.B. gewichteter Mittelpunkt zw. 2 Nachbarn

Unbekannte Zwischenwerte schätzen

Linear Interpolation nicht nice
da viele d.r. was gemischt
→ Glaublich 7,5 ist jetzt nicht

Semantische Segmentierung

Upsampling * 3. Schritt. Rekombination der Klassensets durch
Summierung



Prinzip:

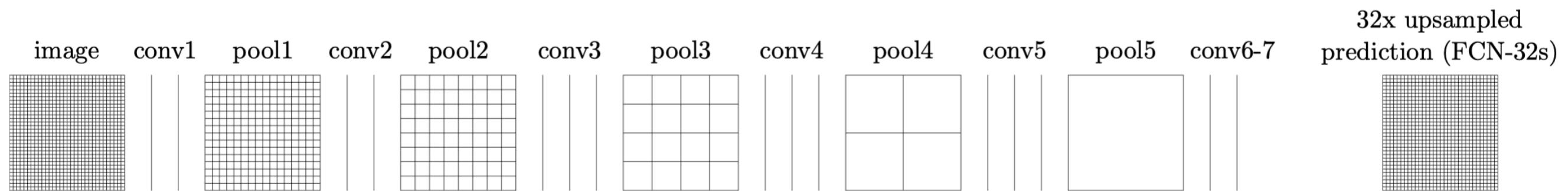
- ↳ pool-4 Ergebnis durch Faltung/Softmax erzielen
- ⇒ Klassenset als Ergebnis in Auflösung von Pool-4
- ↳ Conv-7-Ergebnis um Faktor 2 upsamplen
- ⇒ Klassenset in Auflösung von Pool-4

geht auch mit transponierter Transferring
↓ (gleiche statt fester Interpolation)

© Long et al., 2015

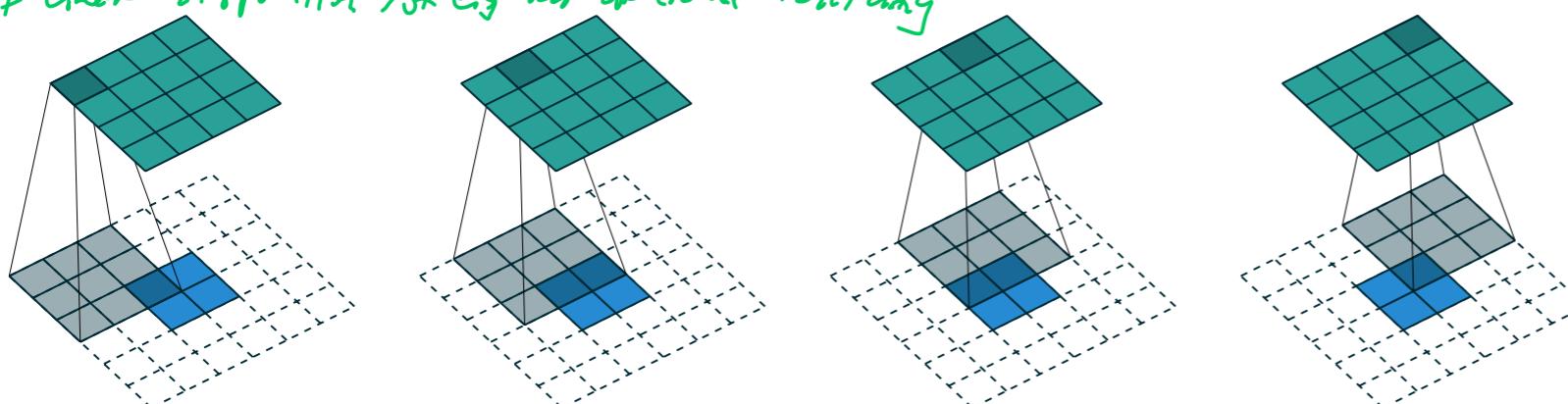
Semantische Segmentierung

Upsampling



Transponierte Faltung:
eig. gelernte Interpolation,
Interpolation wird als Faltung realisiert

heißt Lineare Interpolation ist eig. nur spezielle Faltung



© Dumoulin, Visin

© Long et al., 2015

Einschub: Interpolation

Berechnung eines Wertes „zwischen“ den Pixeln

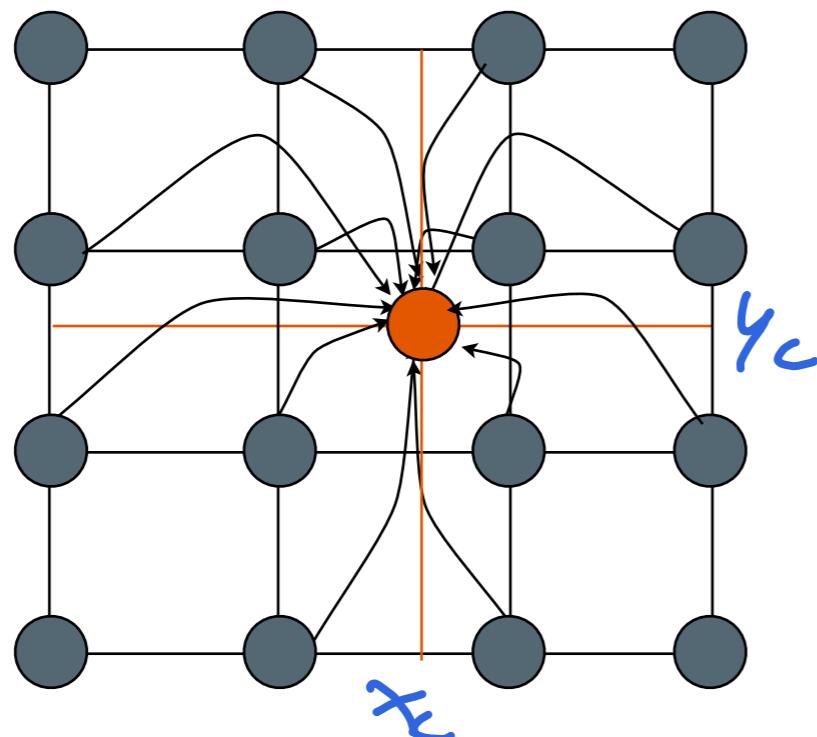
Kontinuierliche Koordinate $(x_c, y_c) \in \mathcal{R}$

Grauwert an der kontinuierlichen Koordinate wird aus den diskreten Nachbarwerten berechnet

Formel ist ne Faltung

$$s(x_c, y_c) = \sum_k \sum_l s(k, l) \cdot h(x_c - k, y_c - l)$$

Interpolationsfunktionskern



Diskrete Positionen (k, l) des bekannten Schriftsatzes

Einschub: Interpolation

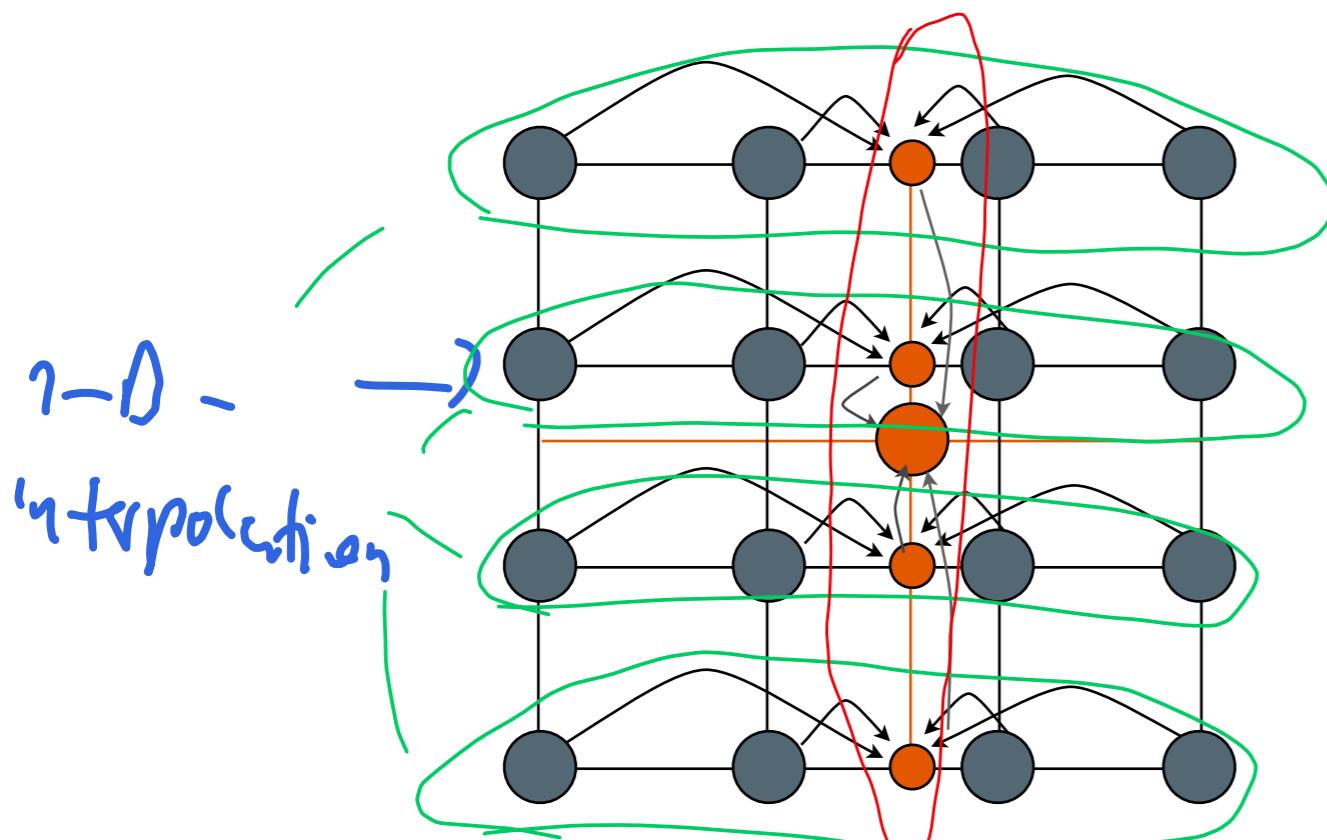
mehrere 1D-Interpolationskerne

Vereinfachung

statt einem 2D-Interpolationskern

$$h(x, y) = h(x) \cdot h(y)$$

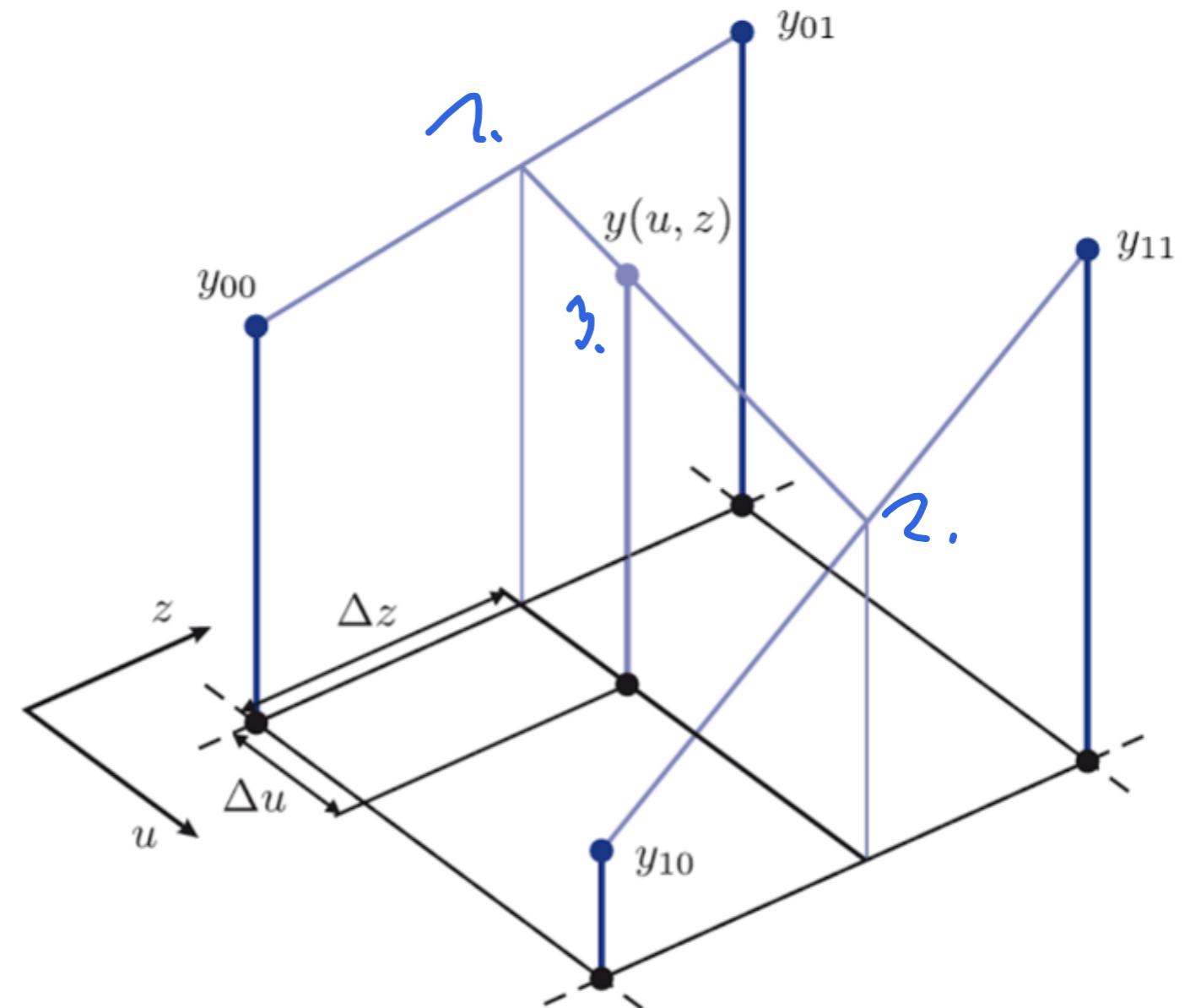
Symmetrisch und separierbar



⇒ Bilinear Interpolation
($2 \times$ Lineare Interpolationen)

Einschub: Interpolation

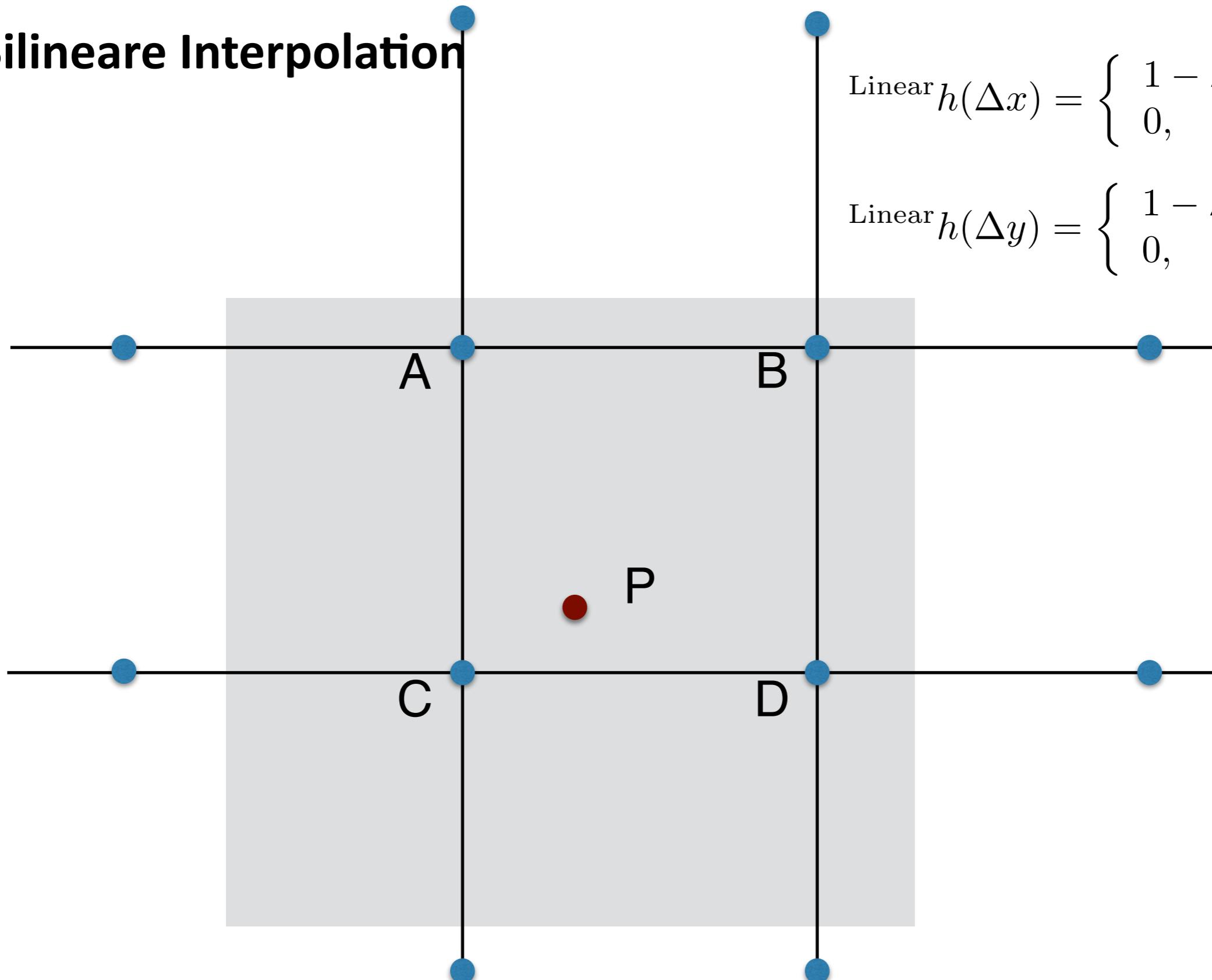
Bilineare Interpolation



aus: Fernando Puerto León: Messtechnik, Systemtheorie für Ingenieure und Informatiker

Einschub: Interpolation

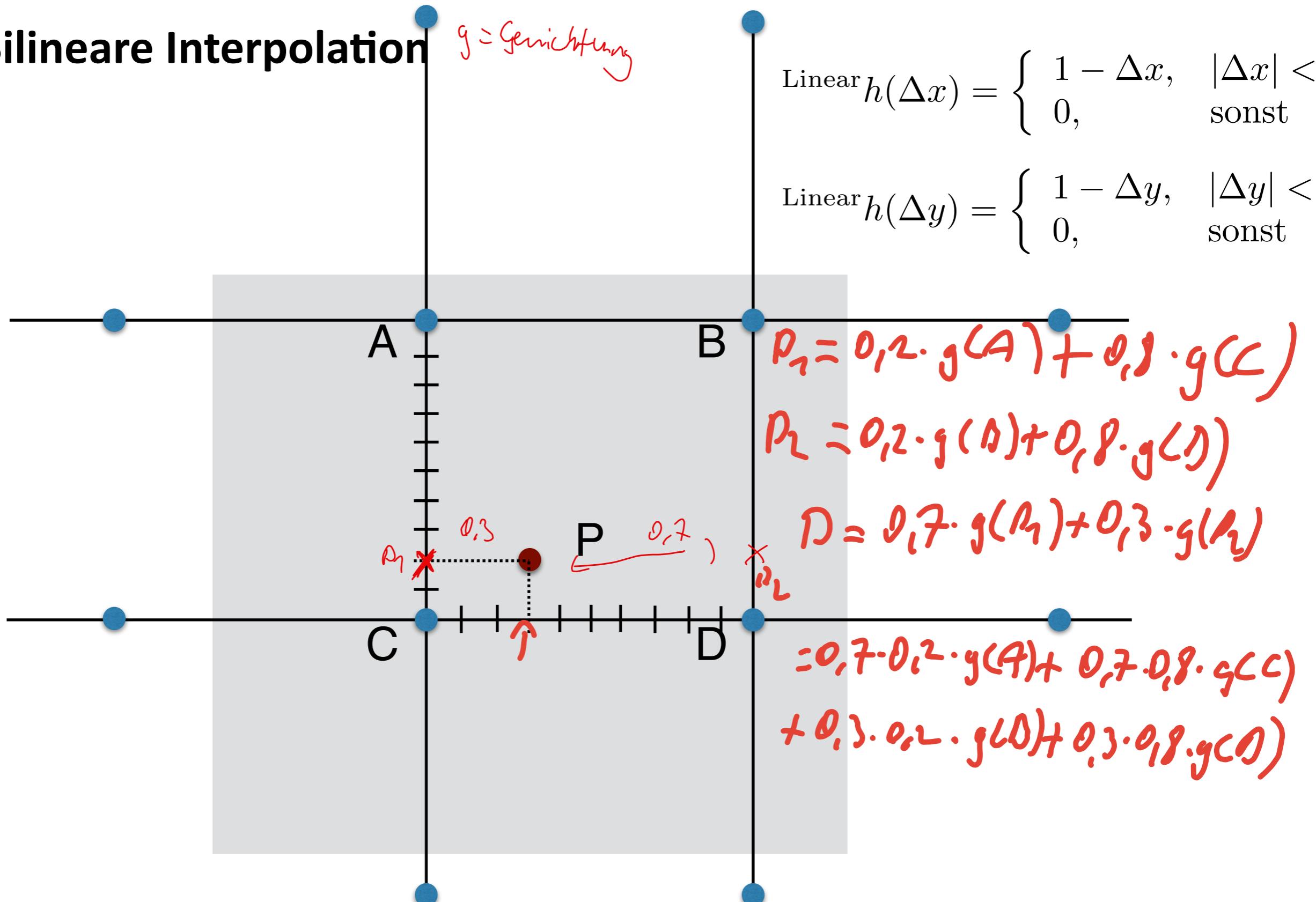
Bilineare Interpolation



Einschub: Interpolation

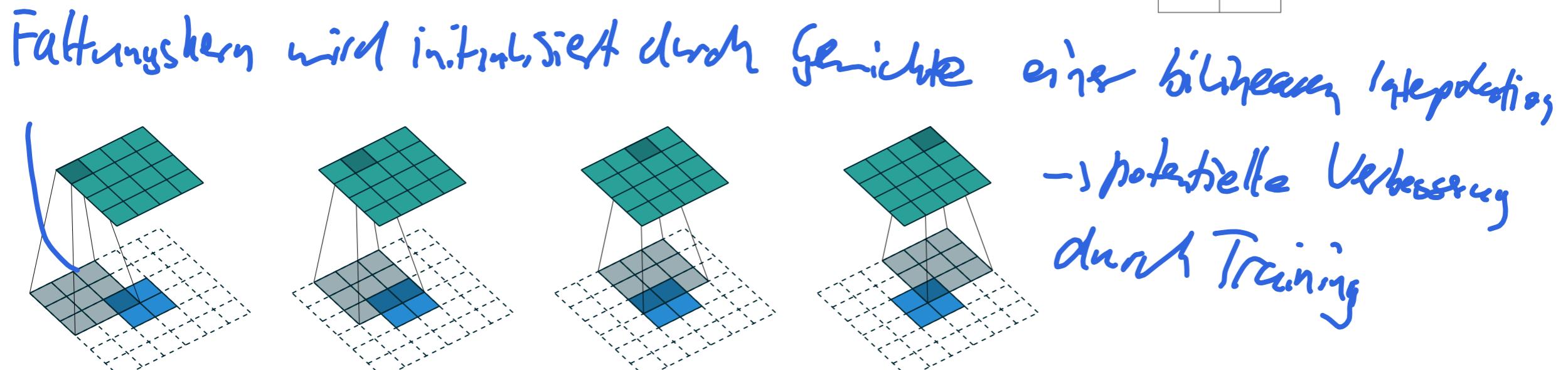
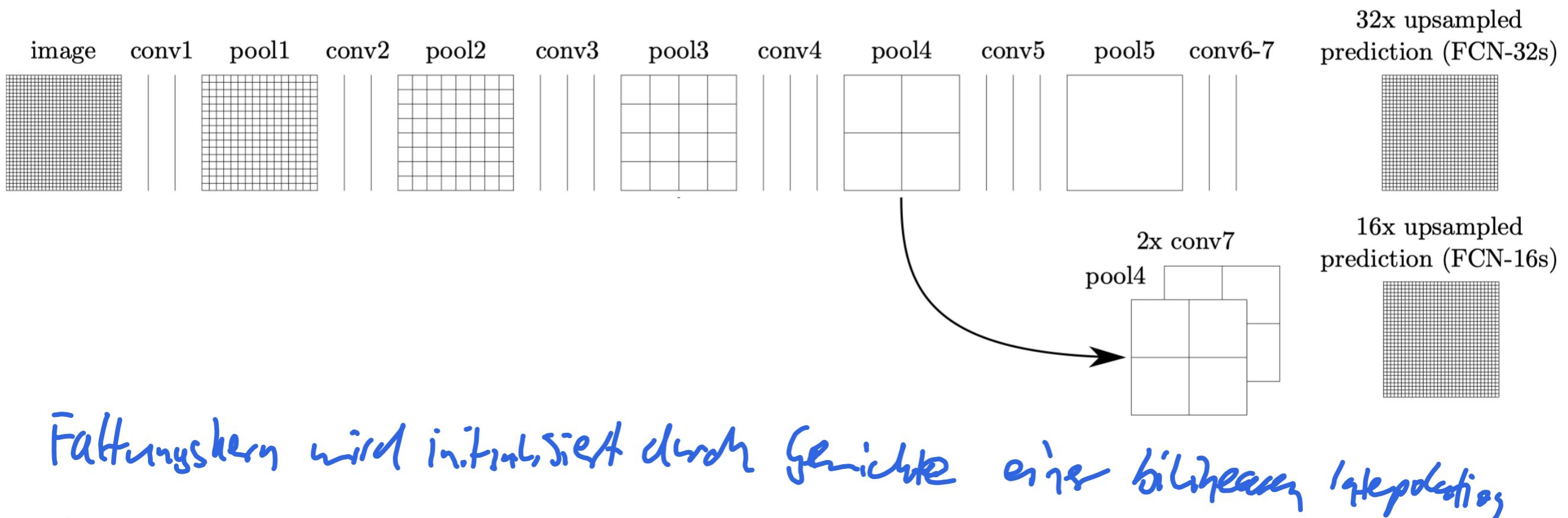
Bilineare Interpolation

g = Gewichtung



Semantische Segmentierung

Upsampling

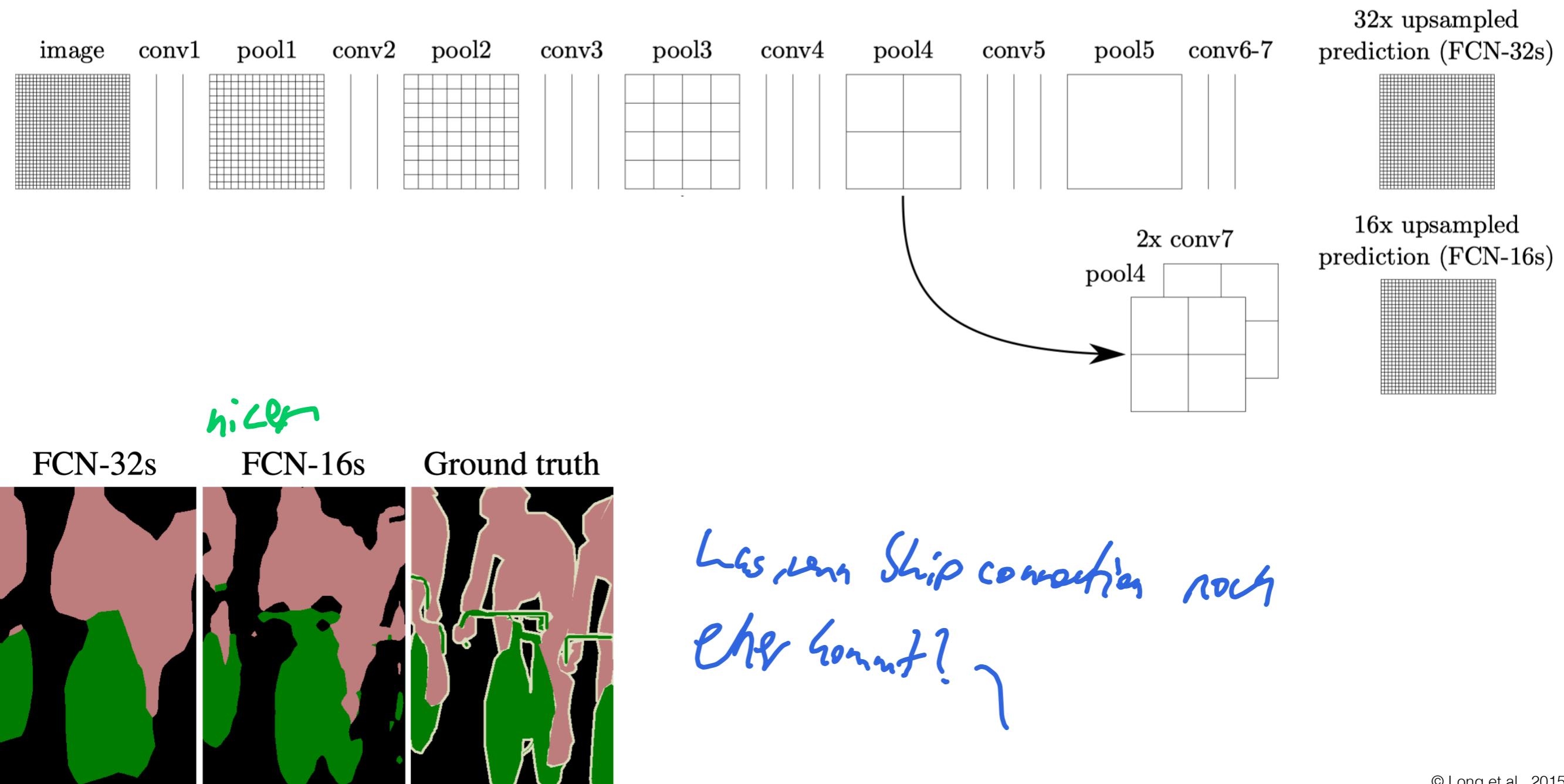


© Dumoulin, Visin

© Long et al., 2015

Semantische Segmentierung

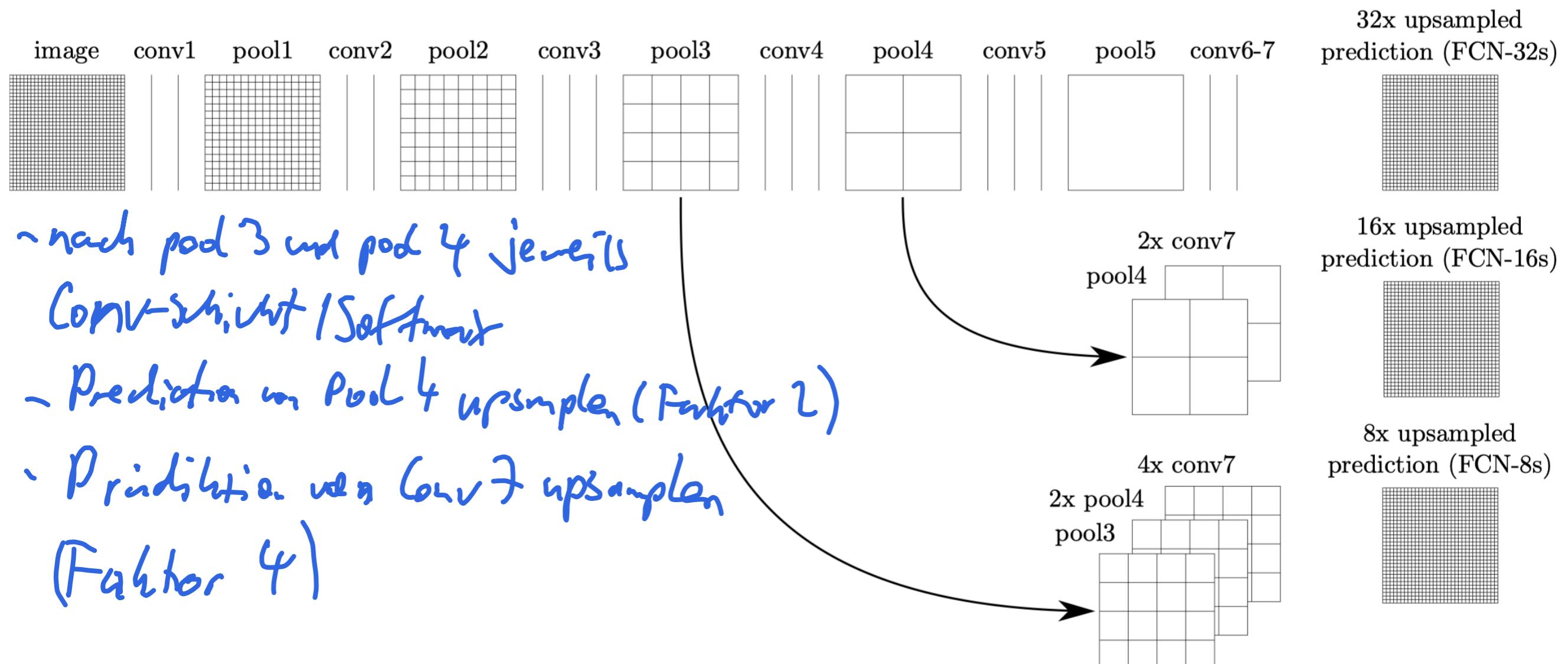
Upsampling



© Long et al., 2015

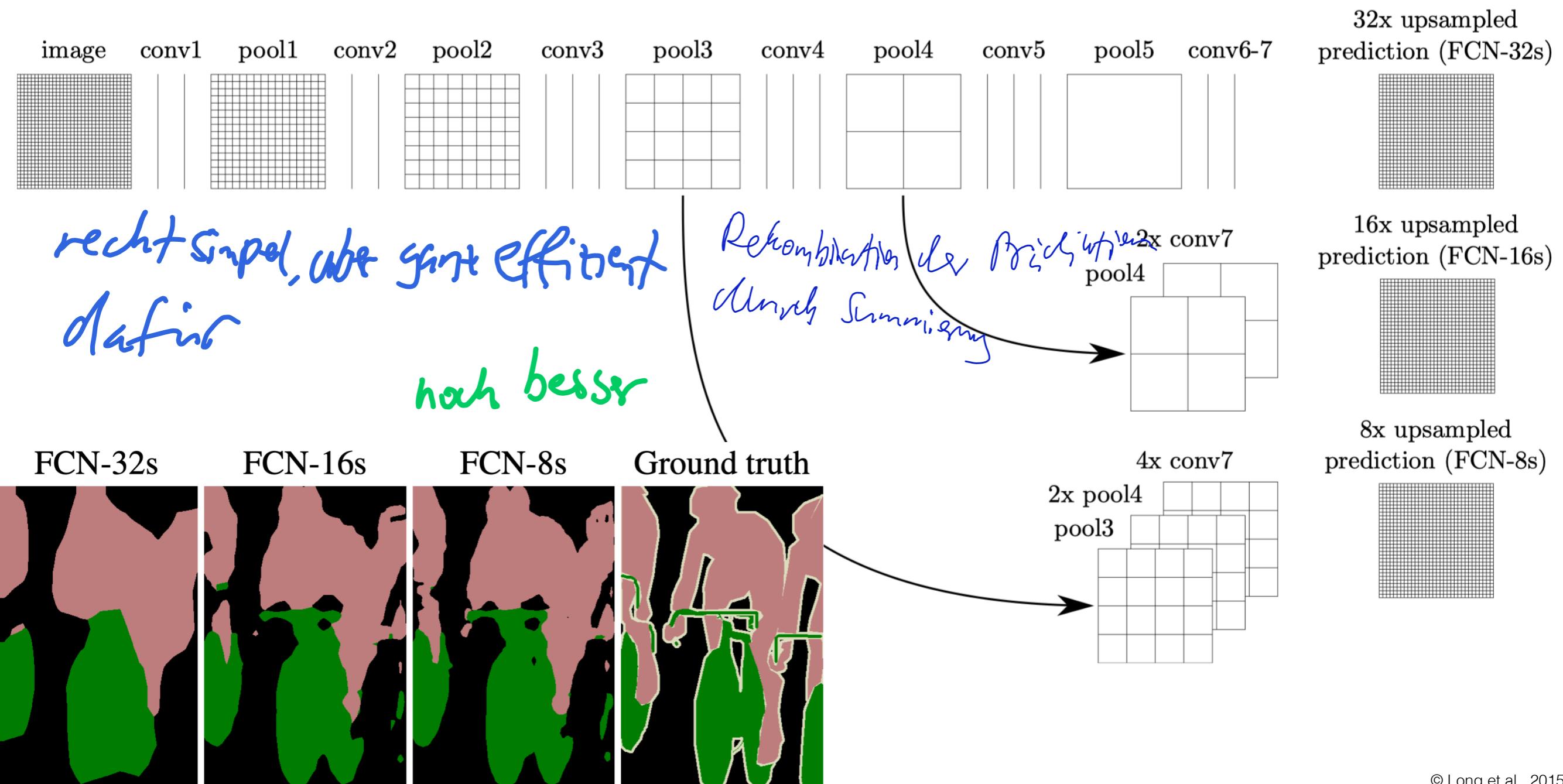
Semantische Segmentierung

Upsampling



Semantische Segmentierung

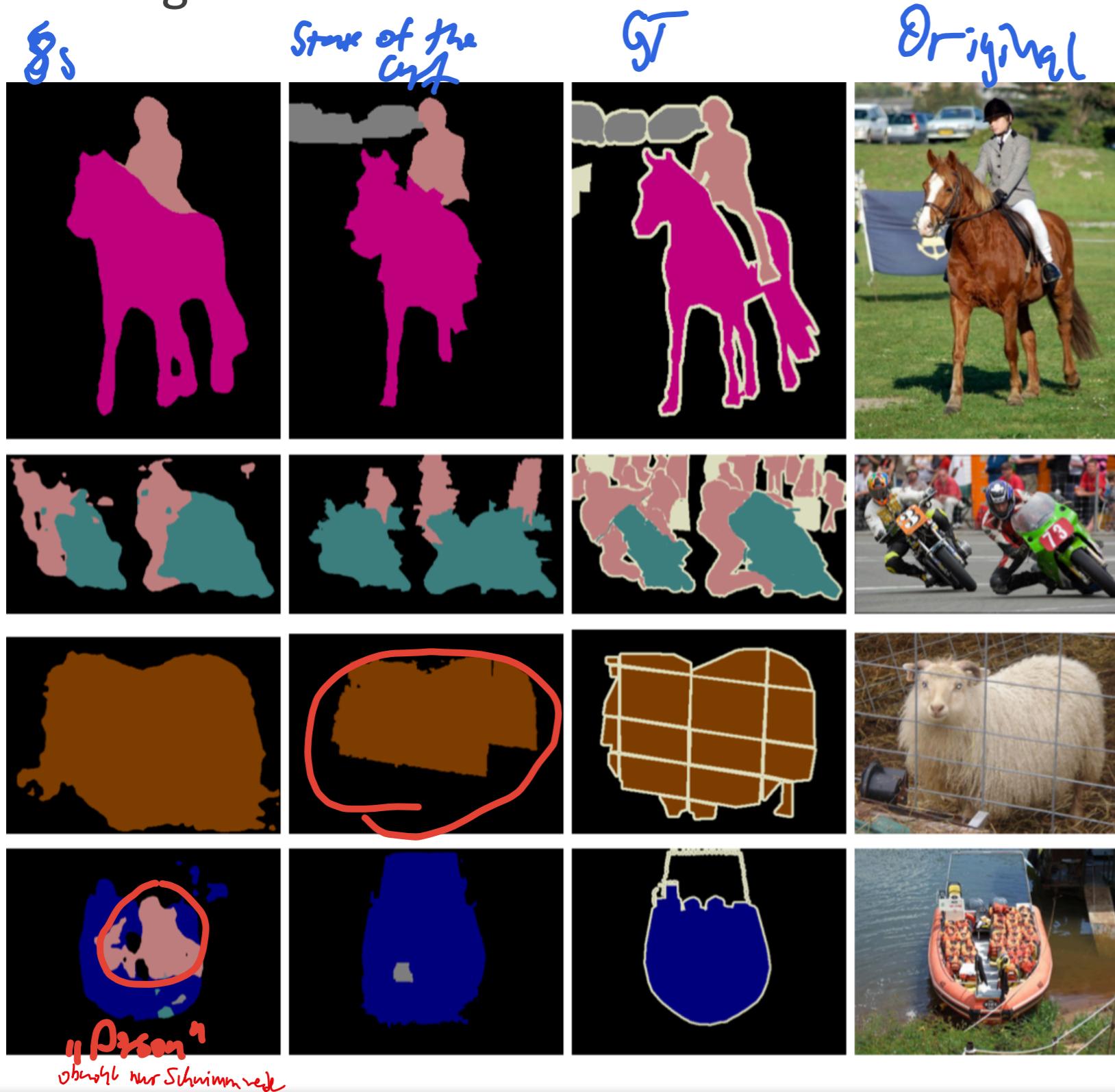
Upsampling



© Long et al., 2015

Semantische Segmentierung

FCN Ergebnisse

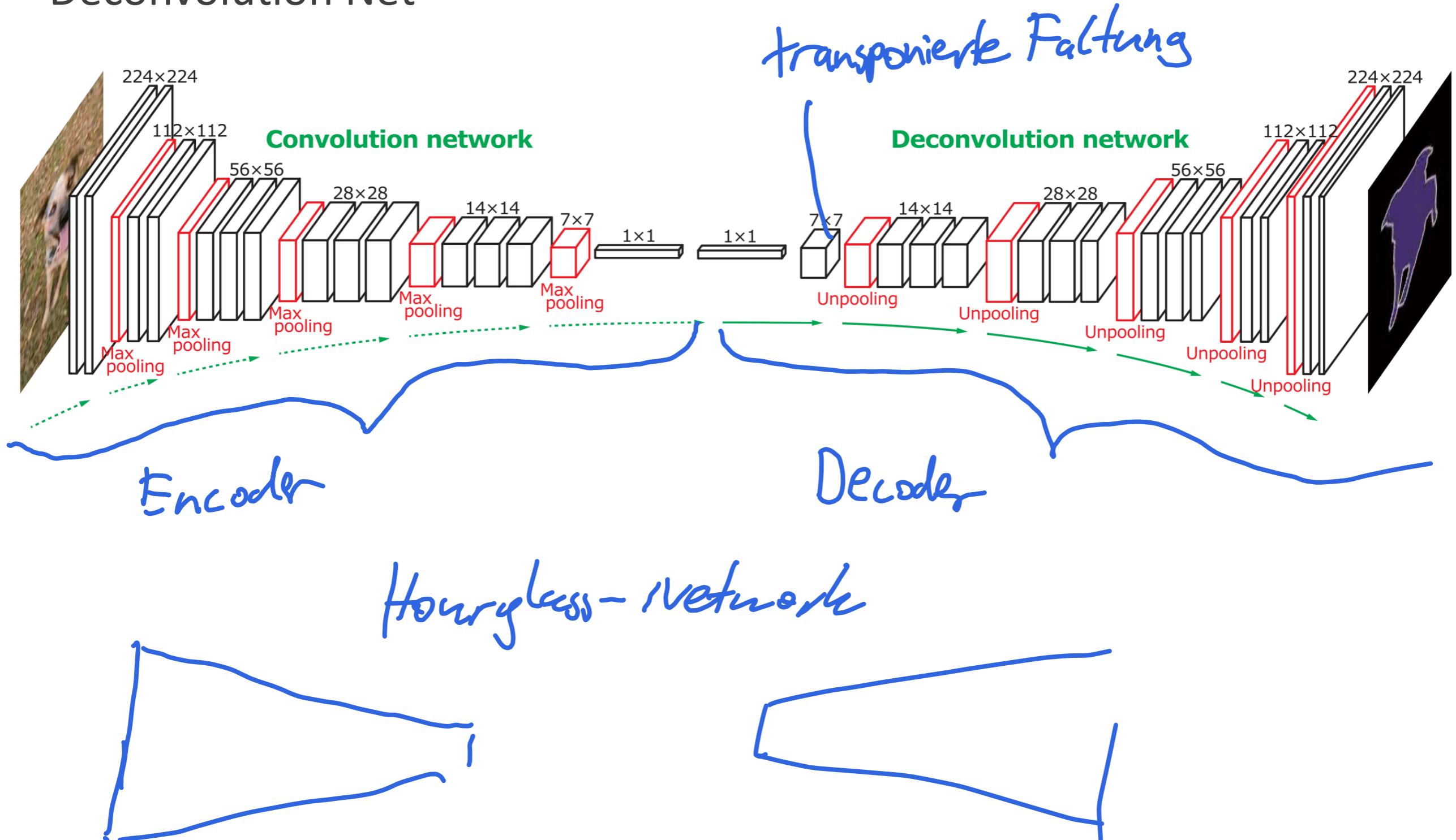


„Person“
obwohl nur Schwimmende

© Long et al., 2015

Semantische Segmentierung

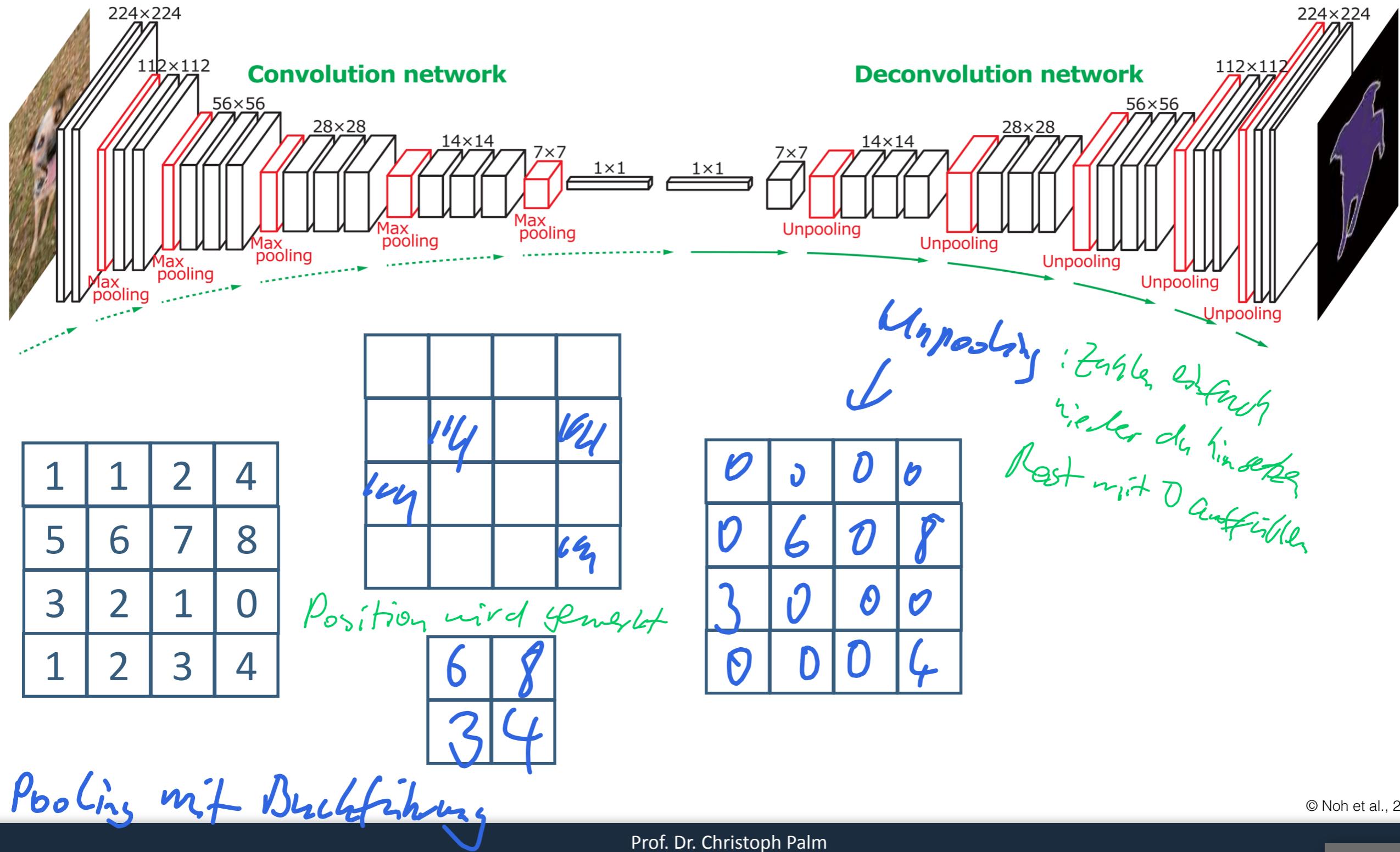
Deconvolution Net



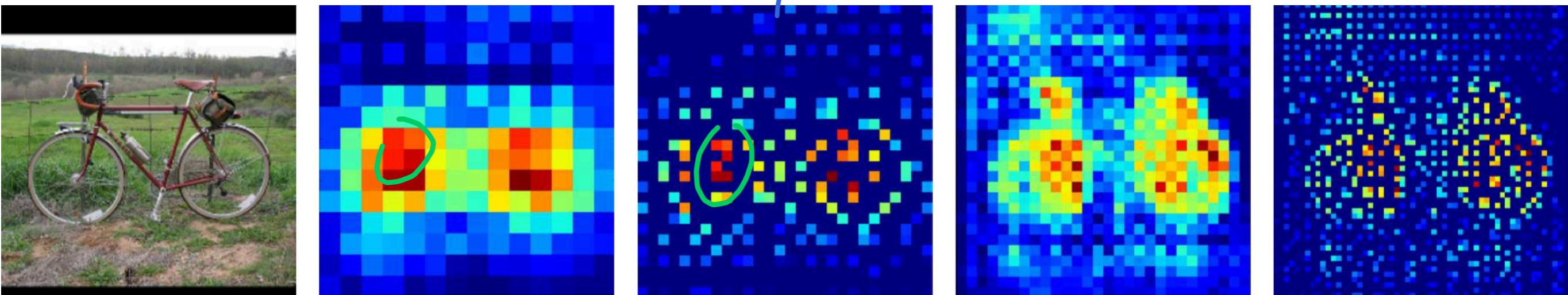
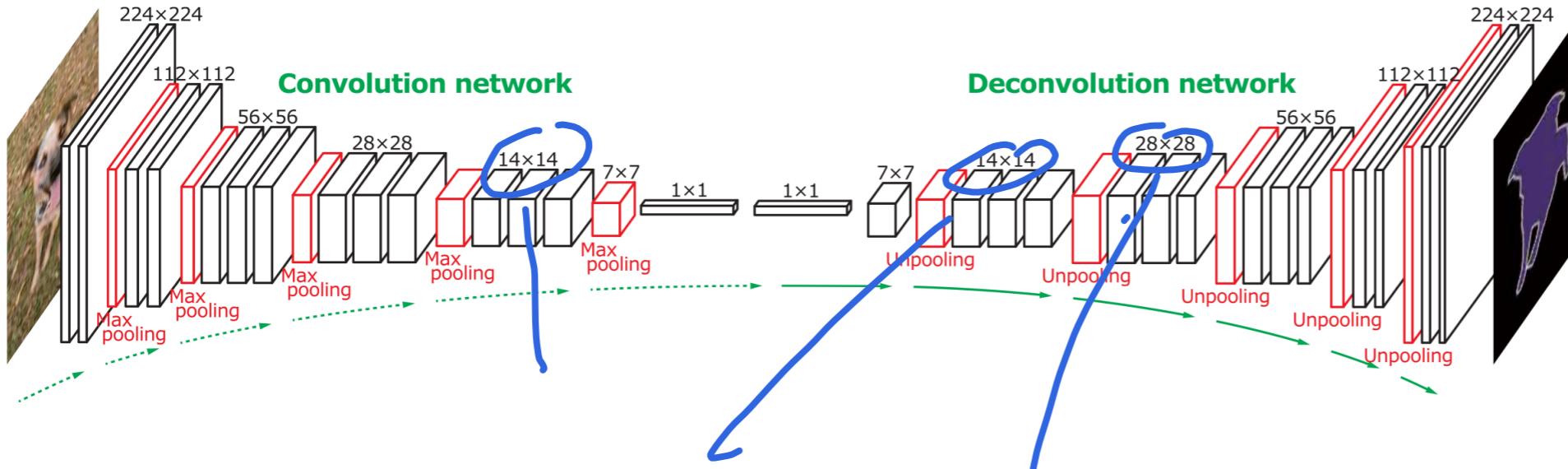
© Noh et al., 2015

Semantische Segmentierung

Deconvolution Net



Semantische Segmentierung

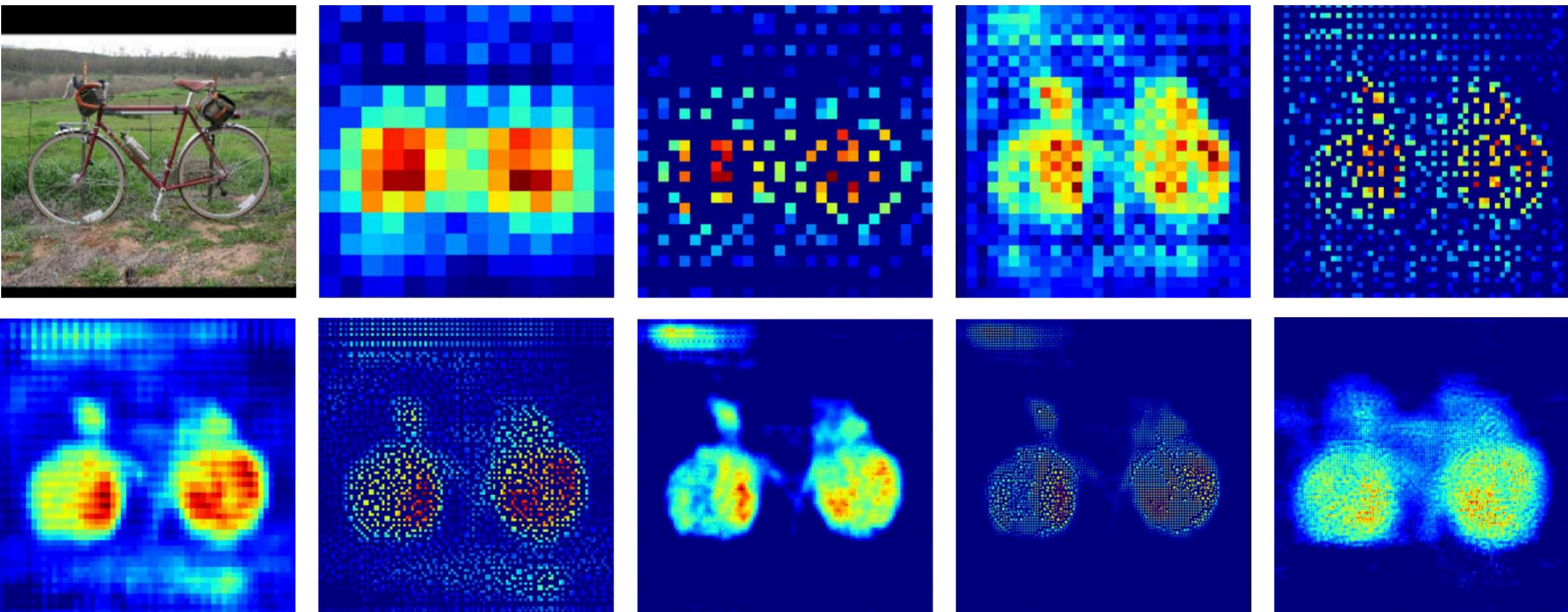
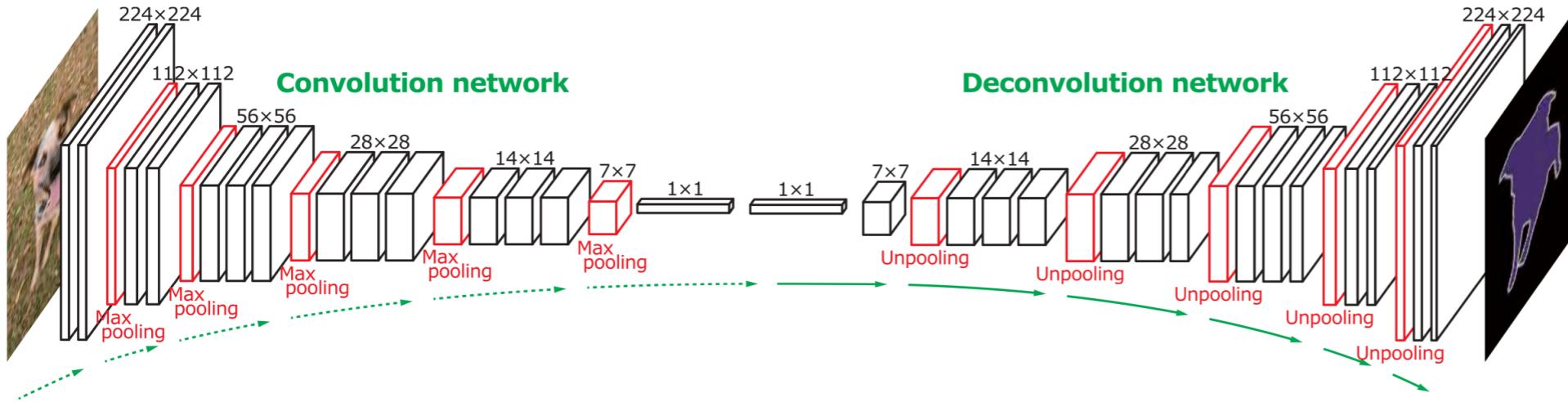


Output
 14×14 Unpooling
 74×74 DeConv

28×28
Unpooling

56×56
Unpooling

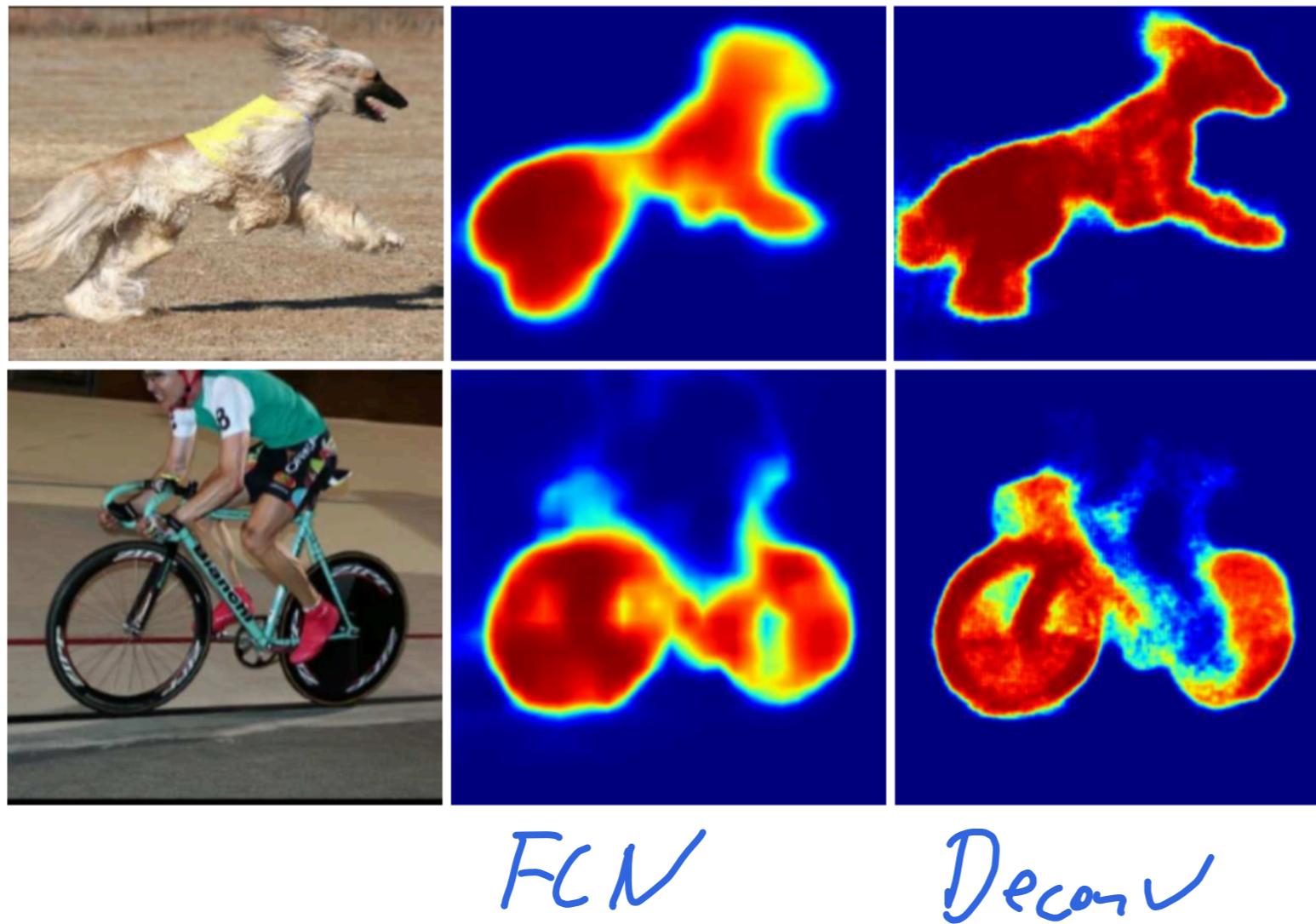
Semantische Segmentierung



© Noh et al., 2015

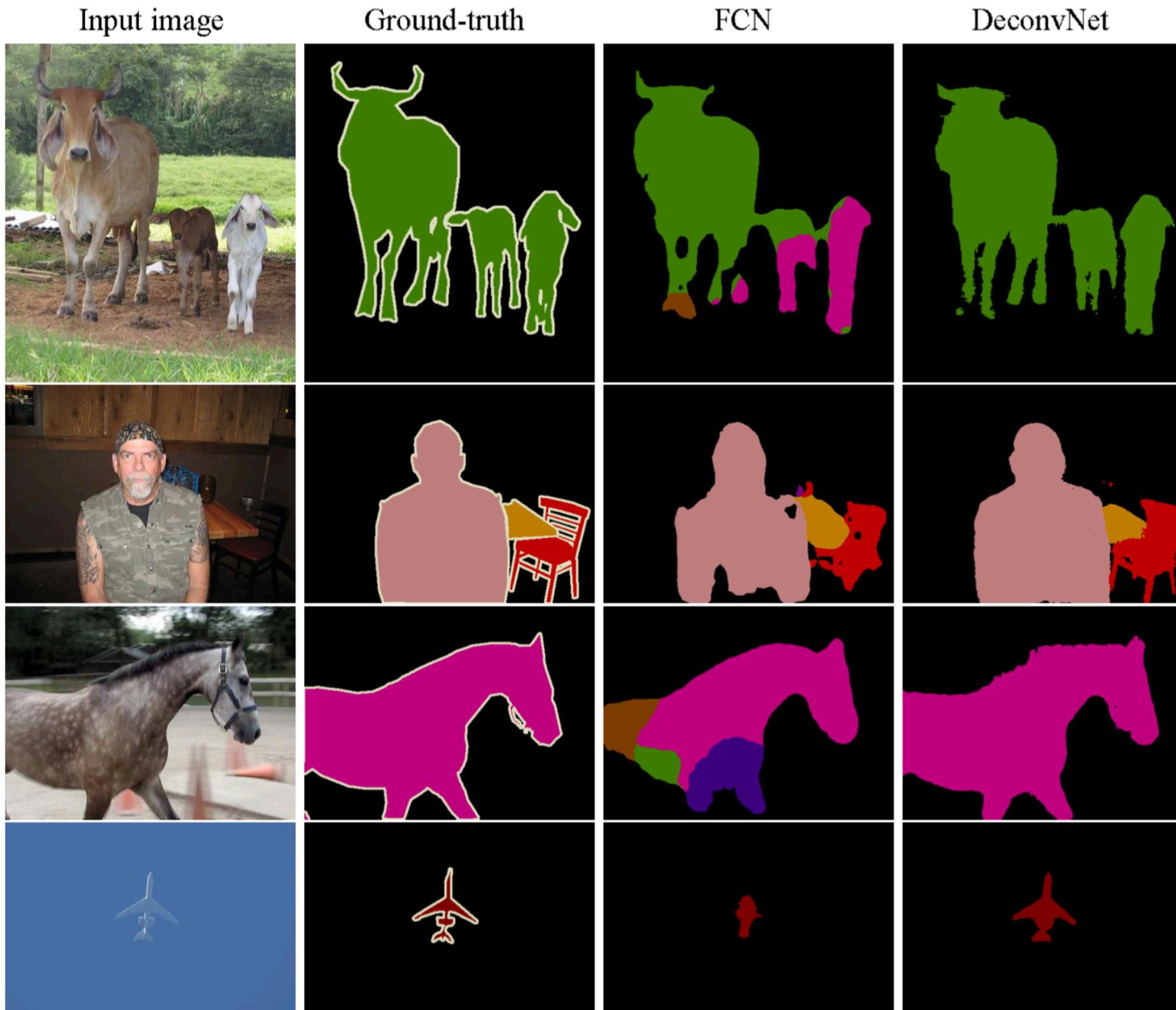
Semantische Segmentierung

Vergleich FCN und Deconvolution Net



Semantische Segmentierung

Vergleich FCN und Deconvolution Net



Deconv hier

besser

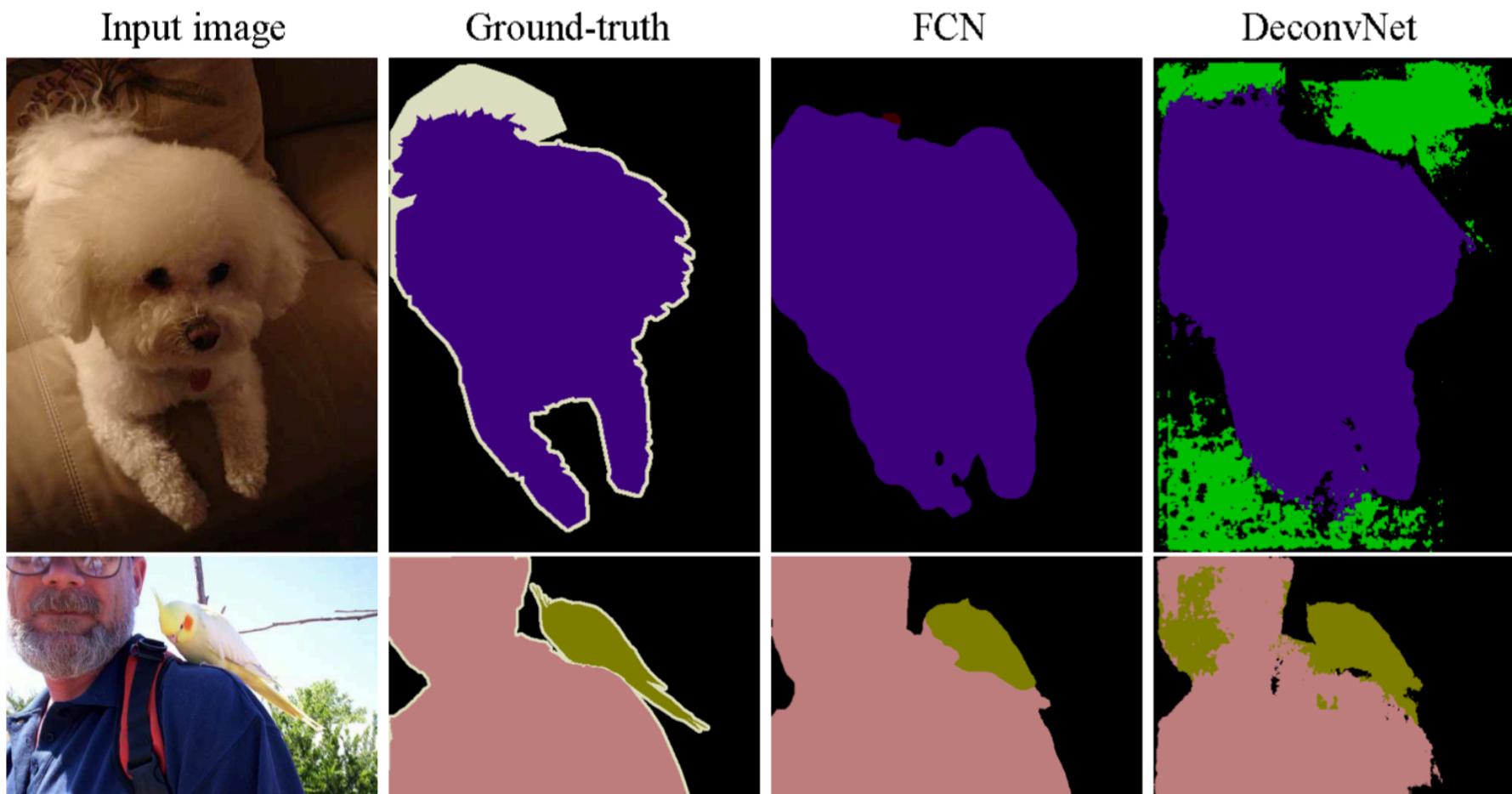
als

FCN

© Noh et al., 2015

Semantische Segmentierung

Vergleich FCN und Deconvolution Net



hier schlechter
als FCN

Semantische Segmentierung

U-Net

≡ Google Scholar

U-net: Convolutional networks for biomedical image segmentation

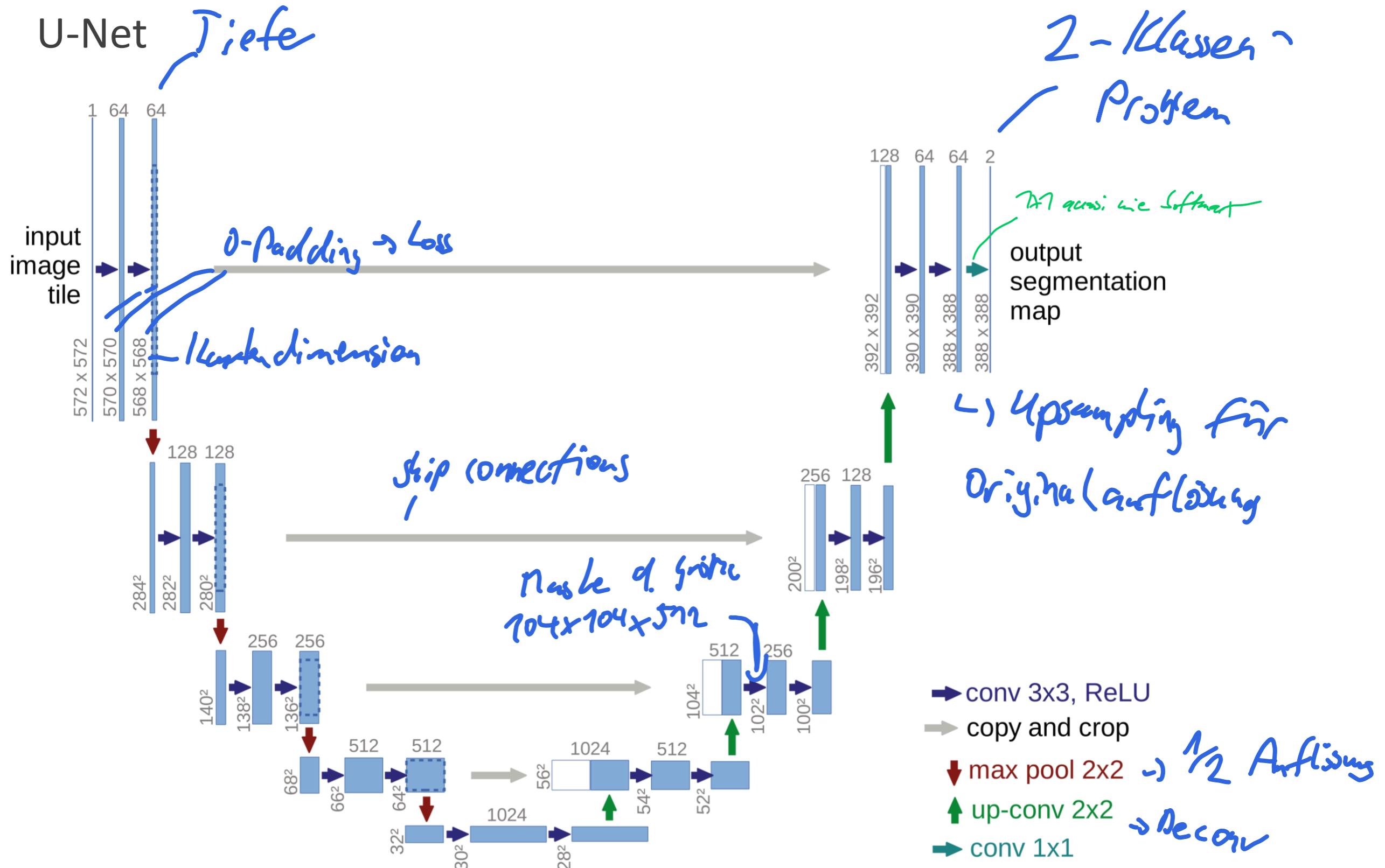
[O Ronneberger, P Fischer, T Brox](#) - International Conference on Medical ..., 2015 - Springer

... We demonstrate the application of the **u-net** to three different segmentation tasks. The first

... The **u-net** (averaged over 7 rotated versions of the input data) achieves without any further ...

☆ Speichern ⚡ Zitieren Zitiert von: 50479 Ähnliche Artikel Alle 30 Versionen In BibTeX importieren

Semantische Segmentierung

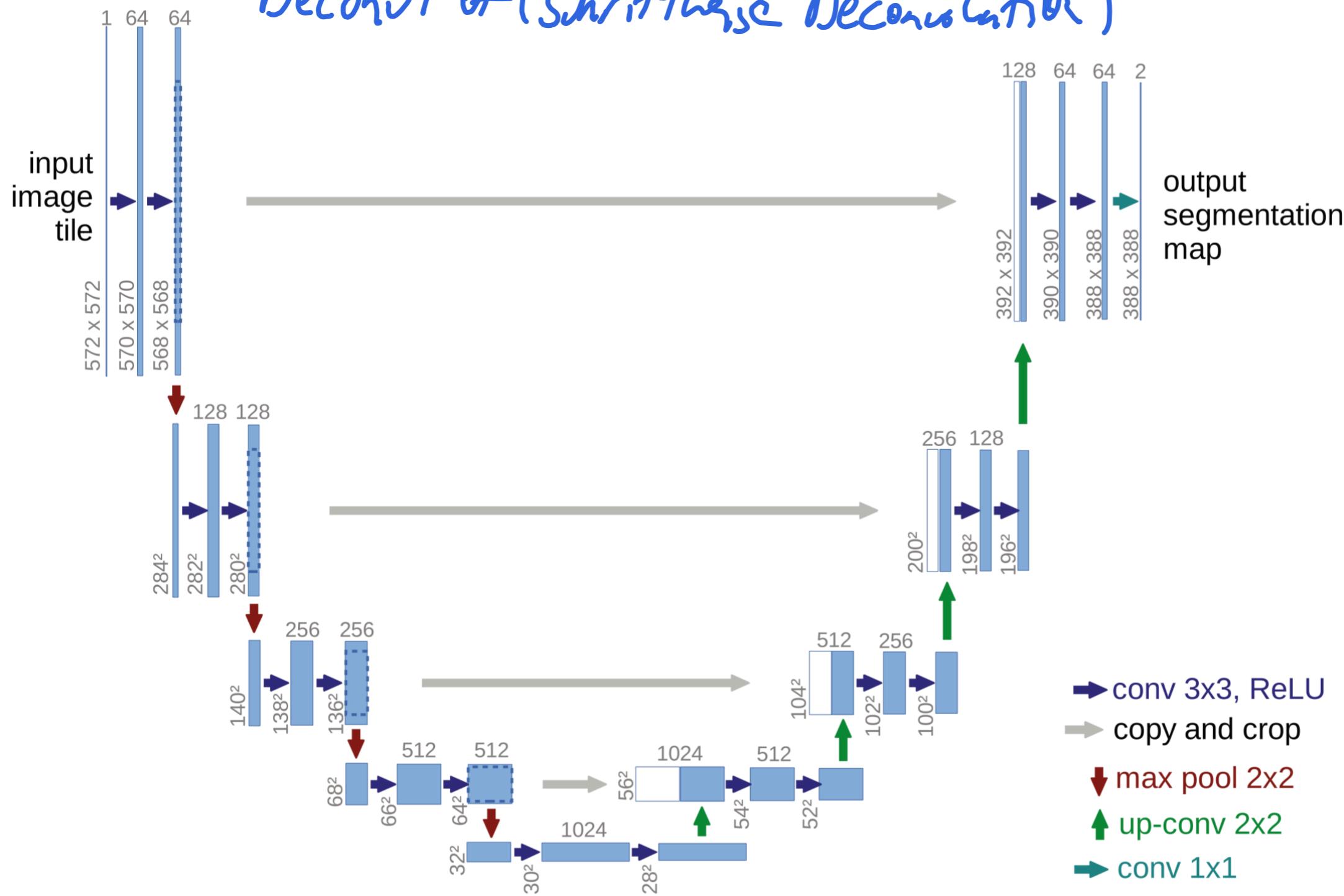


© Ronneberger et al., 2015

Semantische Segmentierung

U-Net

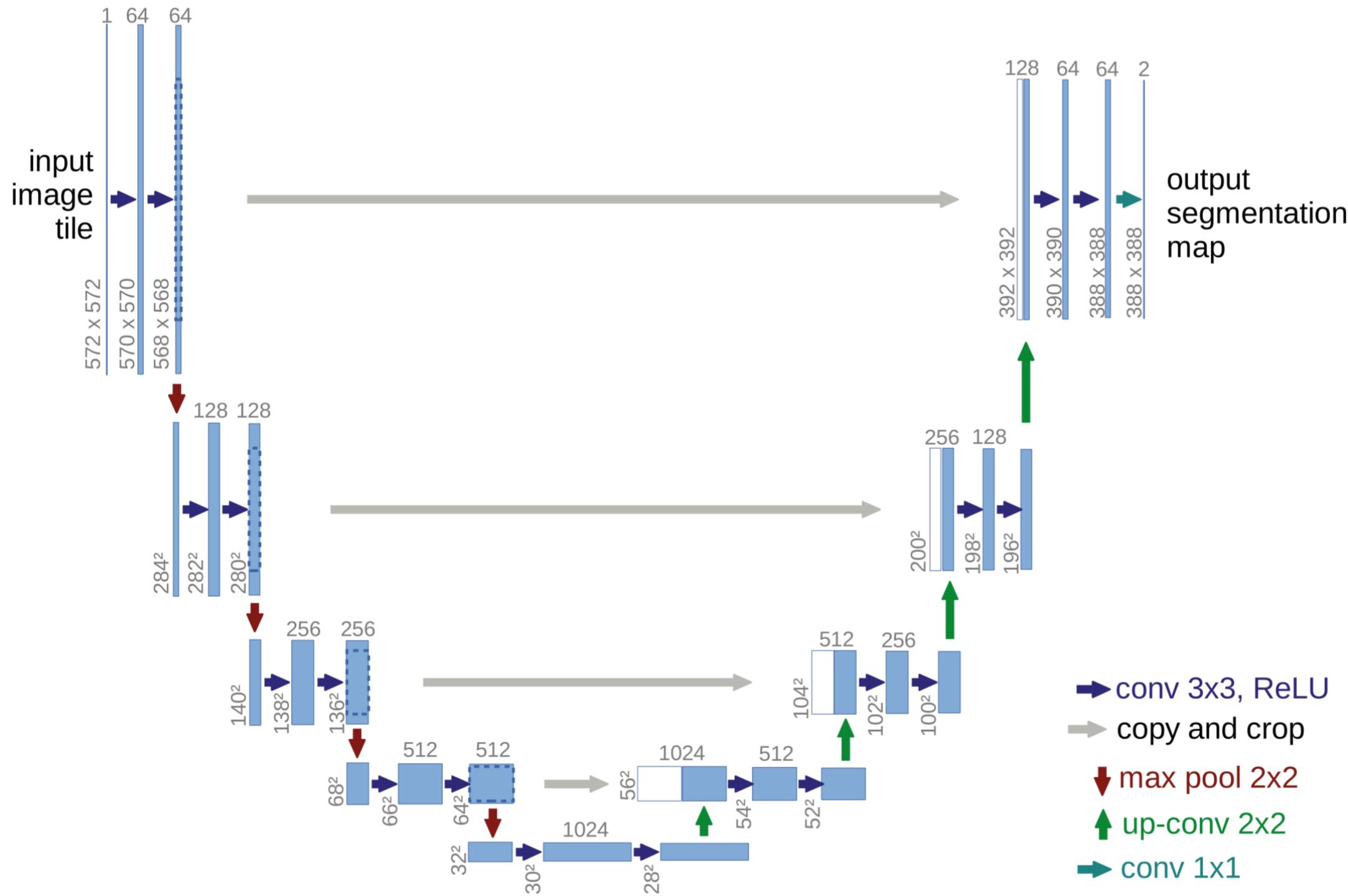
Kombination aus FCN (Skip Connections) und
DeconvNet (schrittweise Deconvolution)



© Ronneberger et al., 2015

Semantische Segmentierung

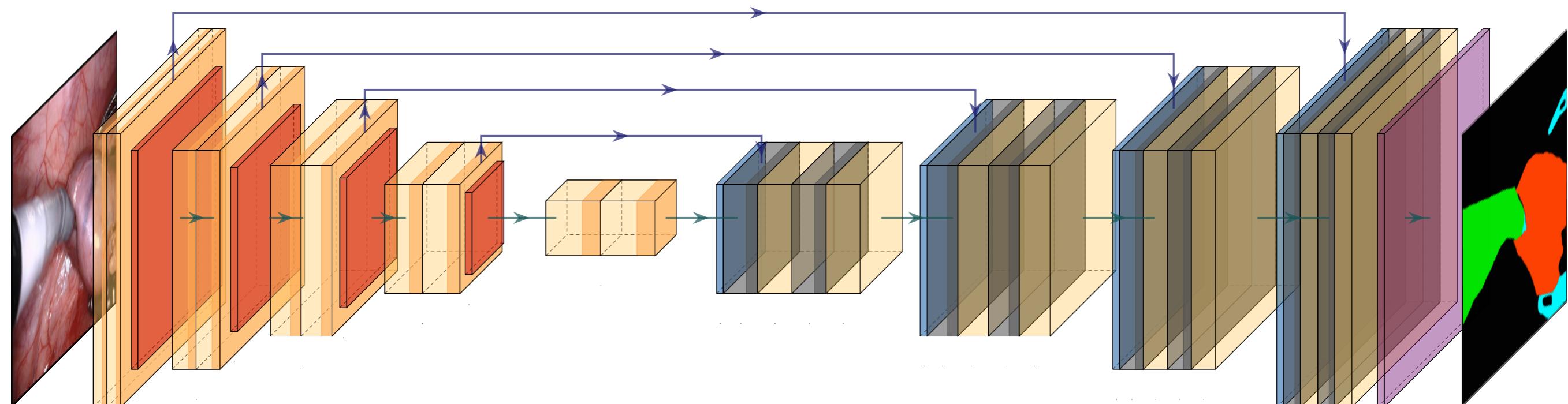
U-Net



© Ronneberger et al., 2015

Semantische Segmentierung

U-Net



© Ronneberger et al., 2015

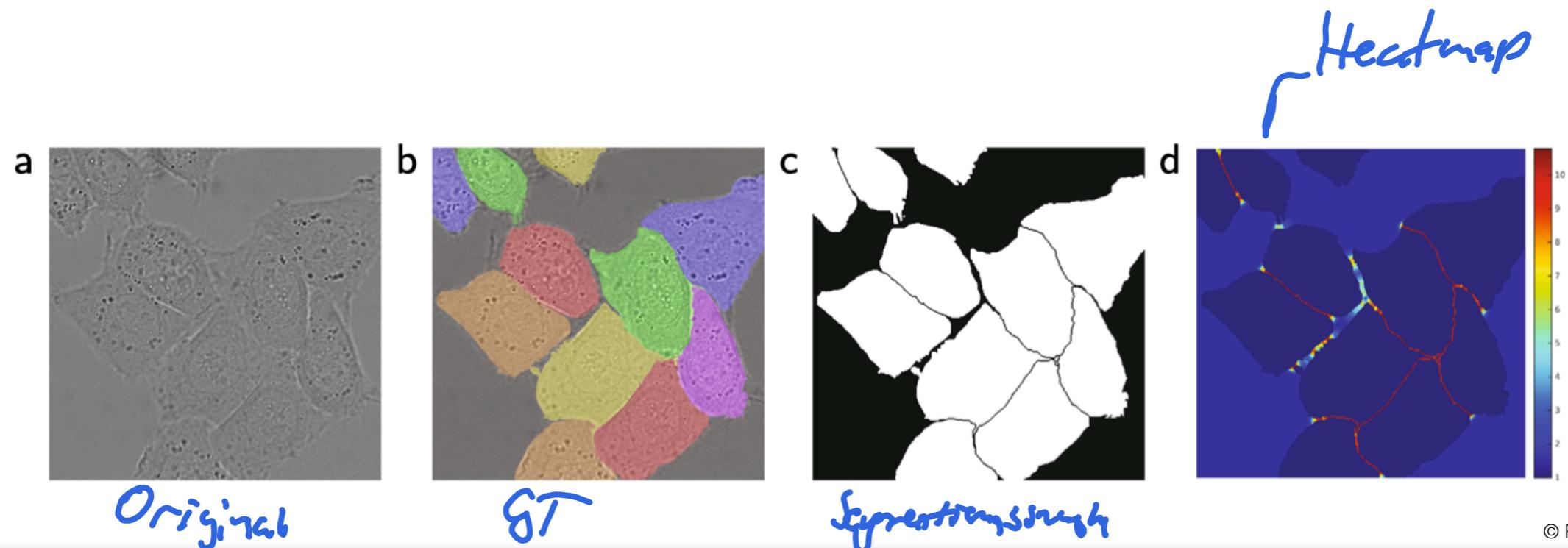
Semantische Segmentierung

U-Net

↳ Patch Size 7

↳ Loss : Cross-Entropy Loss (pixelweise)

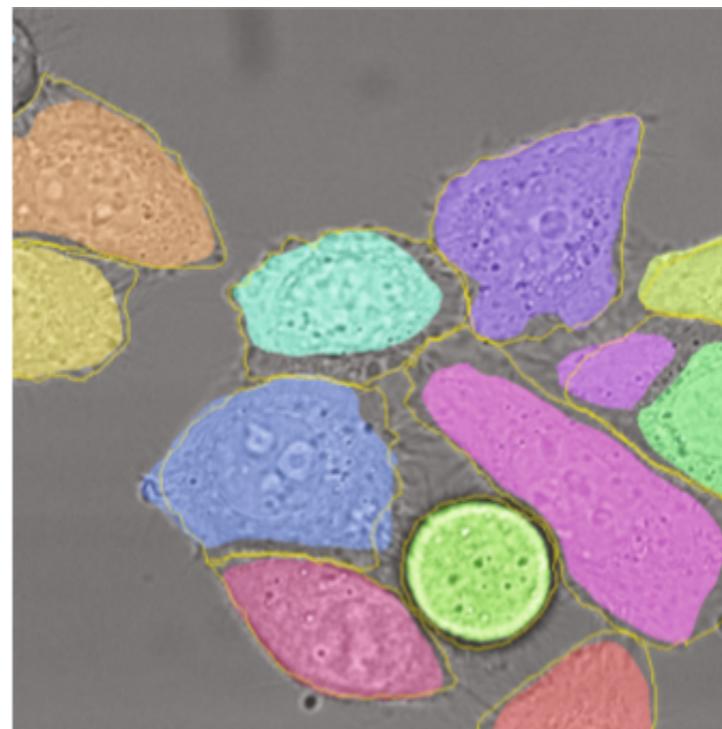
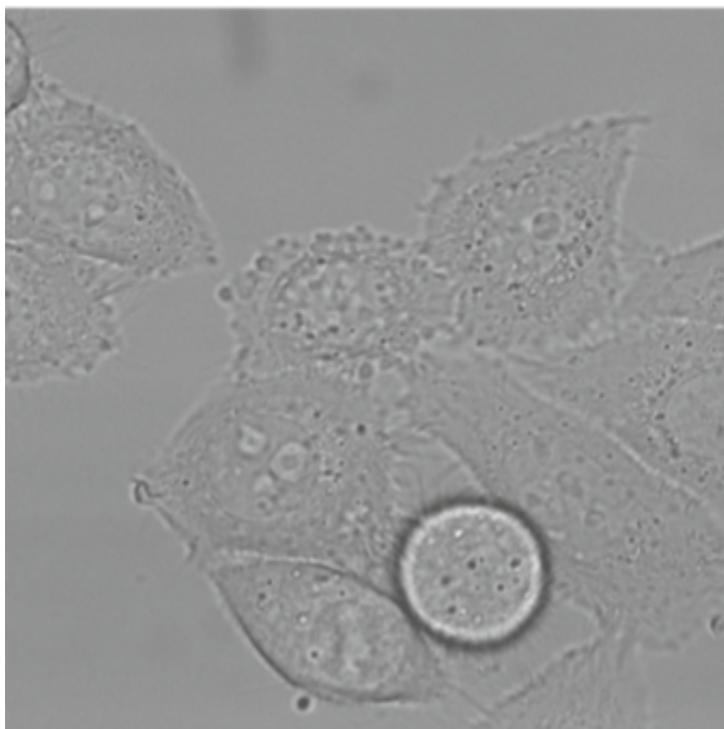
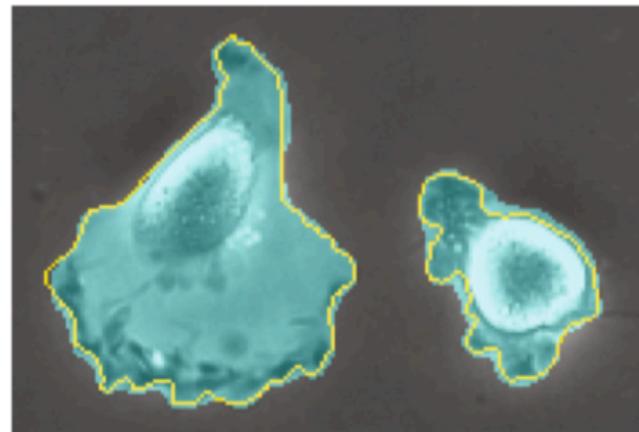
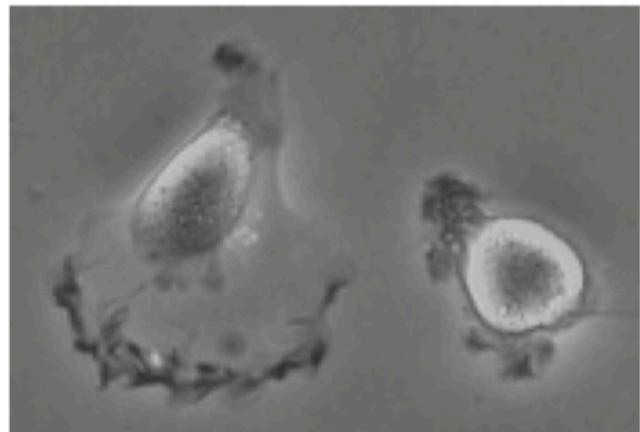
↳ Augmentation : Shift, Rotation, Hellights, Jitter, nicht-lineare Deformation



© Ronneberger et al., 2015

Semantische Segmentierung

U-Net



© Ronneberger et al., 2015

Semantische Segmentierung

nnU-Net no new U-Net

↳ von Fabian Isensee (DUFZ Heidelberg)

⇒ Anpassung von U-Net auf reale Probleme

- Vorbereitung

- Architektur

- Training

ABER: keine echte Veränderung der Architektur

2019: Gewinner der Decathlon-Challenge

⇒ 10 Applikationen, 7 vorab bekannt

⇒ separates Training für jede Applikation

Semantische Segmentierung

nnU-Net

Adaption an Inputgröße

484x512x512 für Leber vs. 36x50x35 für Hippocampus



Größe der Mini-Batches

Architektur: Zahl der Faltungsschichten bzw. der Pooling-Operationen

↳ max: 6 Pooling Operationen

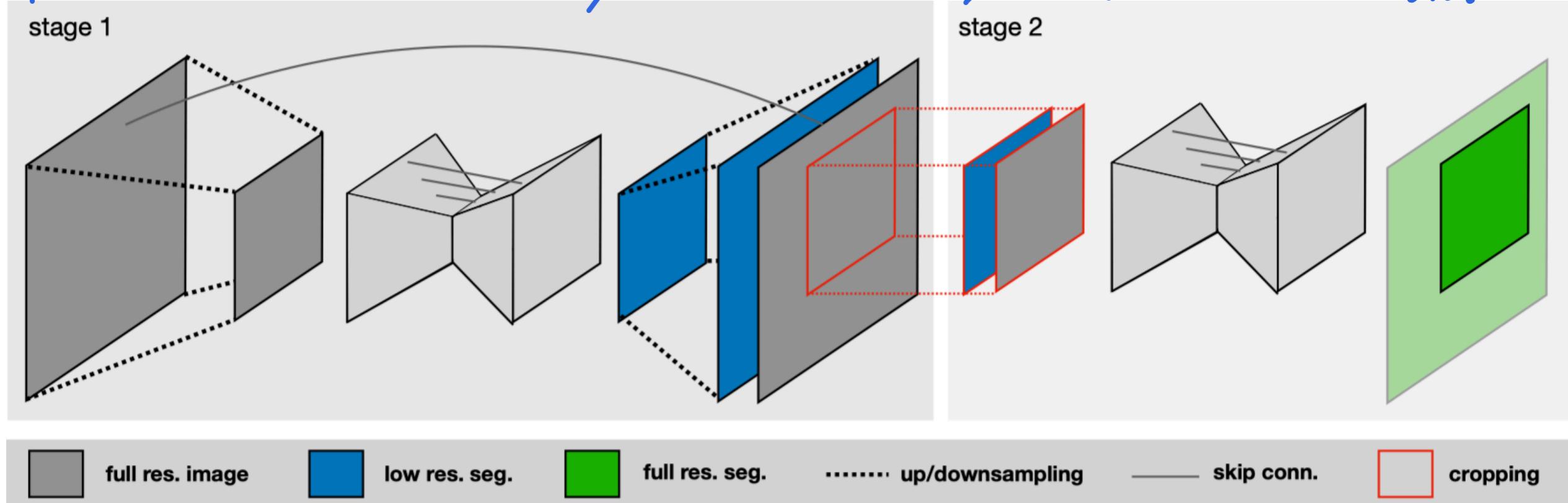
Semantische Segmentierung

nnU-Net

Adaption an Inputgröße

Architektur: U-Net Cascade

für 3D-Daten



© Isensee et al., 2020

Semantische Segmentierung

nnU-Net

Adaption an Inputgröße

Größe der Mini-Batches

Architektur (Zahl der Faltungsschichten, Cascade)

Preprocessing

Crop

Resampling

Normalisierung

auf Bereich ≥ 0

— auf Median-spacing aller Daten

für CT: Extremwerte clippen, dann Normalisieren

\Rightarrow Mittelwert abziehen, durch Standardabweichung teilen

für MRT: patientenspezifisch normalisieren

weil: MRT-Werte sind nicht standardisiert

Semantische Segmentierung

nnU-Net

Adaption an Inputgröße

Größe der Mini-Batches

Architektur (Zahl der Faltungsschichten, Cascade)

Preprocessing

Crop

Resampling

Normalisierung

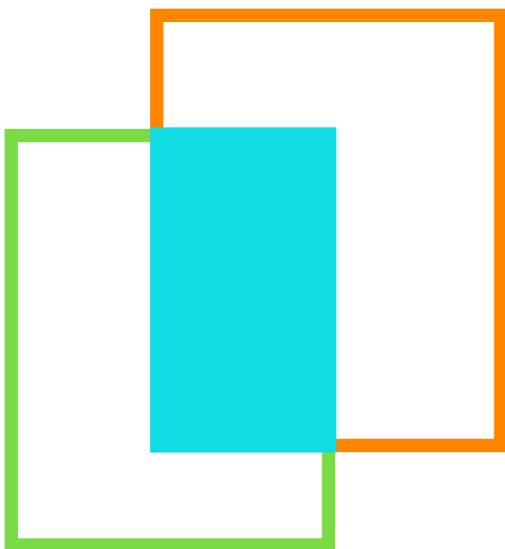
Training

$$\text{Loss: } D = D_{dice} + D_{CE}$$

Dice-Loss
/ \
Cross Entropy-Loss

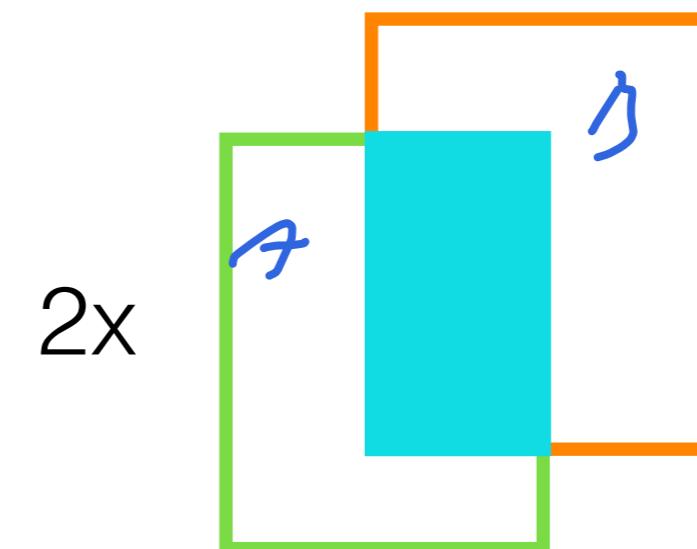
Semantische Segmentierung

Intersection over Union

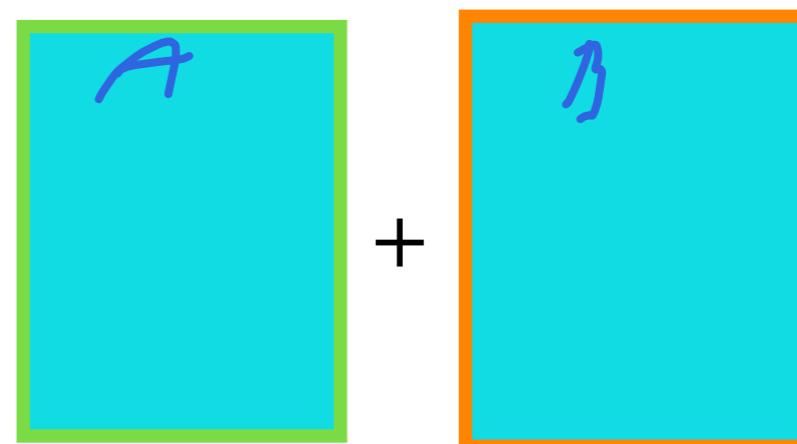


$$\frac{|A \cap B|}{|A \cup B|}$$

Dice



$$\frac{2 \cdot |A \cap B|}{|A| + |B|}$$



→ sind positiv korreliert

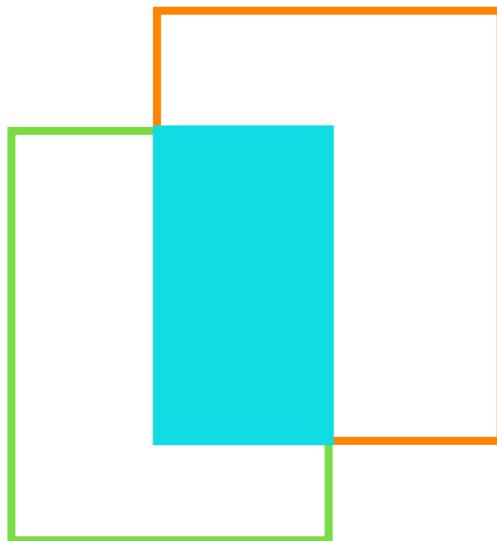
⇒ Dice liefert eher Rücksicht über alle Ergebnisse

⇒ IoT desto stärker einen schlechten Ausweiper

Semantische Segmentierung

Dice-Loss

$$D_{\text{dice}} = \frac{2}{|K|} \sum_{k \in K} \frac{\sum_{i \in I} u_{i,k} v_{i,k}}{\sum_{i \in I} u_{i,k} + \sum_{i \in I} v_{i,k}}$$



$|L|$: Zahl der Klassen
 I : Zahl der Pixel im Trainingsbatch
 u : Softmax-Prediction des Netzes
 v : Ground-Truth; 1-hot-encoded (1 für richtige Klasse,
 0 sonst)



Semantische Segmentierung

nnU-Net

Adaption an Inputgröße

Größe der Mini-Batches

Architektur (Zahl der Faltungsschichten, Cascade)

Preprocessing

Crop

Resampling

Normalisierung

Training

Loss: $D = D_{dice} + D_{CE}$

Mini-Batches:

mindestens eine Vordergrundklasse pro Mini-Batch

Segmentierungsnetzwerke

1. U-Net
2. DeepLab



Einschub: Conditional Random Field

Anwendung: Sequenzklassifikation

$$S(a_k) = \frac{e^{a_k}}{\sum_j e^{a_j}} \quad \text{mit } a_k = \sum_{j=0}^{P_{l-1}} \hat{w}_{kj}^l \cdot z_j^{l-1}$$

und l ist letzte Logit-Schicht

$$p(\hat{y}^{(t)} | \boldsymbol{x}^{(t)}; \hat{w}) \quad \begin{array}{l} \text{a-posteriori: } \mathcal{V}_k \\ \text{(Ergebnis eines NNs)} \end{array}$$



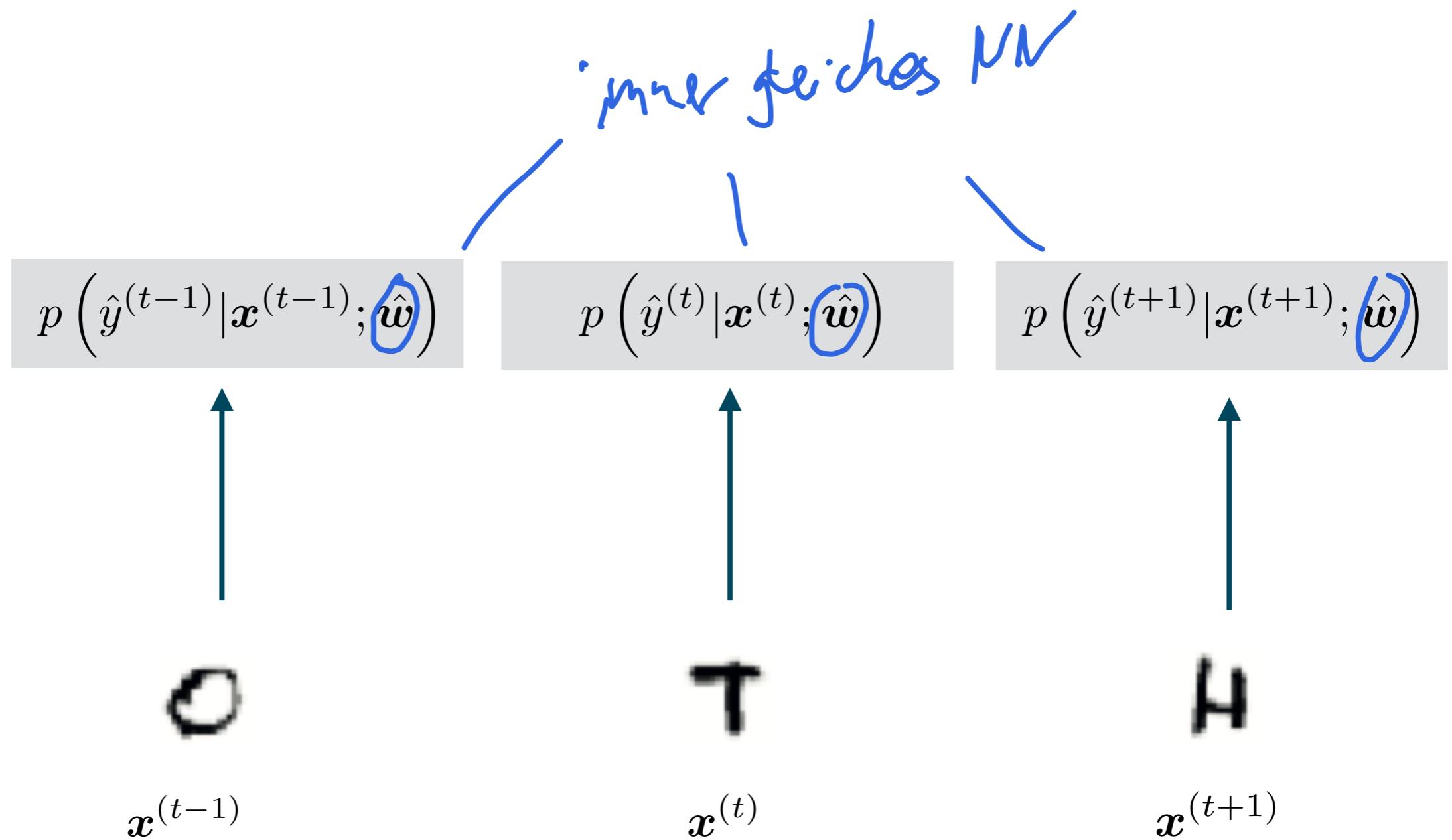
T

$\boldsymbol{x}^{(t)}$

© H. Larochelle

Einschub: Conditional Random Field

Anwendung: Sequenzklassifikation

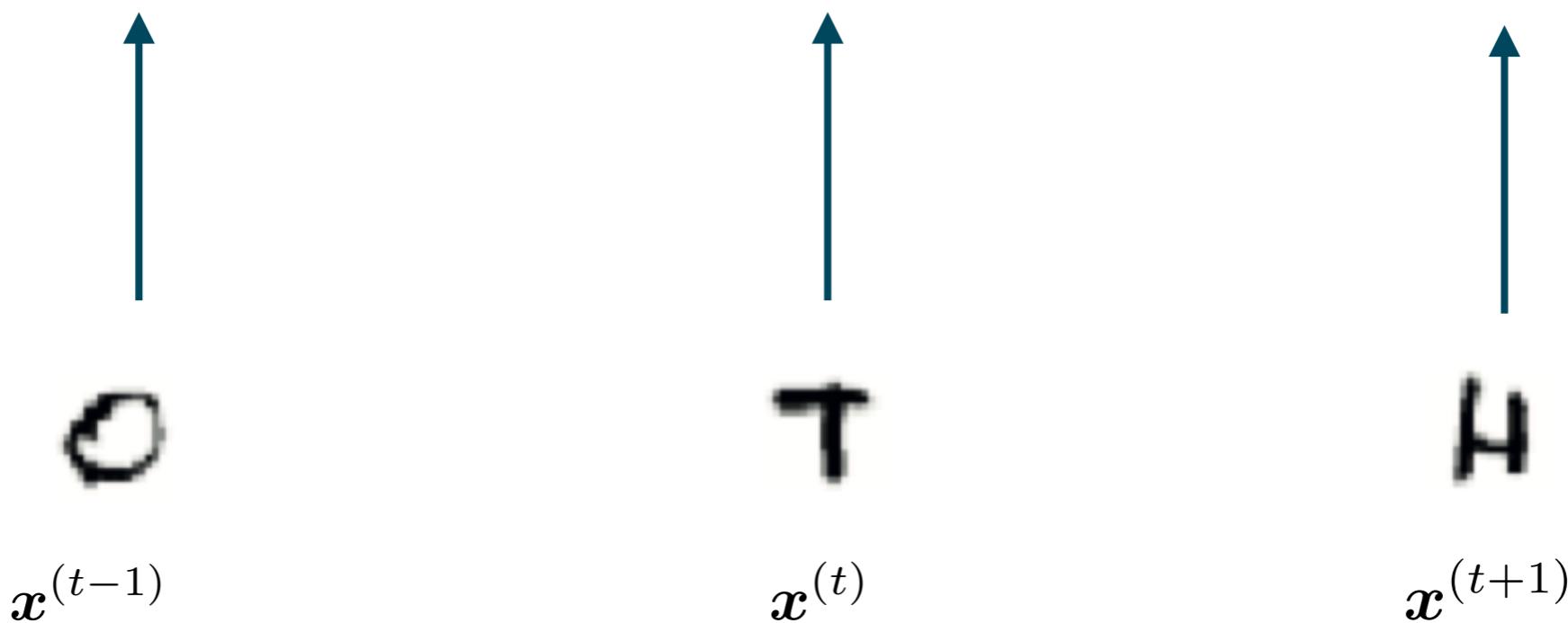


Einschub: Conditional Random Field

Anwendung: Sequenzklassifikation

✓ einer Sequenz von Klassen, geben eine
Sequenz von Beobachtungen

$$p \left(\hat{y}^{(1)}, \dots, \hat{y}^{(T)} | \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(T)}; \hat{\mathbf{w}} \right)$$



© H. Larochelle

Einschub: Conditional Random Field

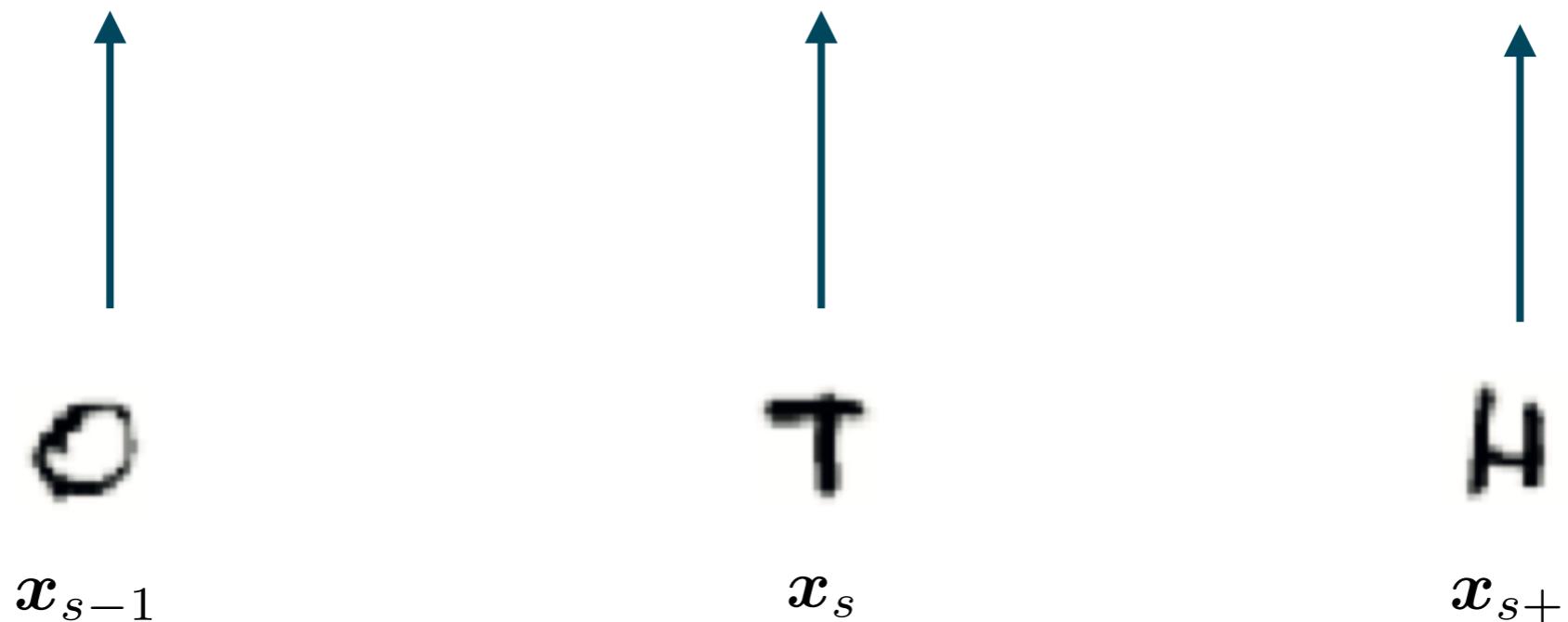
Anwendung: Sequenzklassifikation

$$\mathbf{X}^{(t)} = \left[\mathbf{x}_1^{(t)}, \dots, \mathbf{x}_s^{(t)}, \dots, \mathbf{x}_{S_t}^{(t)} \right]$$

*S: Länge der Sequenz
s: Index*

$$\mathbf{y}^{(t)} = \left[y_1^{(t)}, \dots, y_s^{(t)}, \dots, y_{S_t}^{(t)} \right]$$

$$p(\mathbf{y}|\mathbf{X}) = p(y_1, \dots, y_S | \mathbf{x}_1, \dots, \mathbf{x}_S)$$



© H. Larochelle

Einschub: Conditional Random Field

Anwendung: Sequenzklassifikation

Klassifikation

$$= \prod_s p(y_s | \mathbf{x}_s)$$

$$= \prod_s \left(e^{a_{y_s}(\mathbf{x}_s)} / \sum_j e^{a_{y_j}(\mathbf{x}_s)} \right) - \text{Softmax einsetzen}$$

$$p(\mathbf{y} | \mathbf{X}) = p(y_1, \dots, y_S | \mathbf{x}_1, \dots, \mathbf{x}_S)$$

a_{y_s} : ausgenutzt für Klasse y_s
(Logit)

Einschub: Conditional Random Field

Anwendung: Sequenzklassifikation

Klassifikation

$$= \prod_s p(y_s | \mathbf{x}_s)$$

$$= \exp \left(\sum_s a_{y_s}(\mathbf{x}_s) \right) / \left(\prod_s Z(\mathbf{x}_s) \right)$$

Variable für
Nenner einsetzen

$$p(\mathbf{y} | \mathbf{X}) = p(y_1, \dots, y_S | \mathbf{x}_1, \dots, \mathbf{x}_S)$$

Einschub: Conditional Random Field

Anwendung: Sequenzklassifikation

Klassifikation

$$= \prod_s p(y_s | \mathbf{x}_s)$$

$$= \exp \left(\sum_s a_{y_s}(\mathbf{x}_s) \right) / \left(\prod_s Z(\mathbf{x}_s) \right)$$

$$p(\mathbf{y} | \mathbf{X}) = p(y_1, \dots, y_S | \mathbf{x}_1, \dots, \mathbf{x}_S)$$

Sequenzklassifikation: linearer Zusammenhang

$$= \exp \left(\sum_{s=1}^S a_{y_s}(\mathbf{x}_s) + \sum_{s=1}^{S-1} V_{y_s, y_{s+1}} \right) / Z(\mathbf{X})$$

W_h ges. 1 Bedeutung
W_h für Segm_t

Matrix mit W_h
V-Matrix
Klasse 3, Klasse 5 → Zeile 3, Spalte 5 liefert
W_h für Segm_t aus 3 und 5

→ quadratischer Anfang

→ linear durch dynamische Programmierung

© H. Larochelle

Einschub: Conditional Random Field

Anwendung: Sequenzklassifikation

Kontextfenster

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}) &= \exp \left(\sum_{s=1}^S a_{y_s}(\mathbf{x}_s) + \sum_{s=1}^{S-1} V_{y_s, y_{s+1}} \right. \\ &\quad \left. + \sum_{s=2}^S a_{y_s}(\mathbf{x}_{s-1}) + \sum_{s=1}^{S-1} a_{y_s}(\mathbf{x}_{s+1}) \right) / Z(\mathbf{X}) \end{aligned}$$

Einschub: Conditional Random Field

Anwendung: Sequenzklassifikation

Kontextfenster

$$p(\mathbf{y}|\mathbf{X}) = \exp \left(\sum_{s=1}^S a_{y_s}(\mathbf{x}_{s-1}, \mathbf{x}_s, \mathbf{x}_{s+1}) + \sum_{s=1}^{S-1} V_{y_s, y_{s+1}} \right) / Z(\mathbf{X})$$

$$= \exp \left(\sum_{s=1}^S a_u(y_s) + \sum_{s=1}^{S-1} a_p(y_s, y_{s+1}) \right) / Z(\mathbf{X})$$

© H. Larochelle

Einschub: Conditional Random Field

Notation über Faktoren:

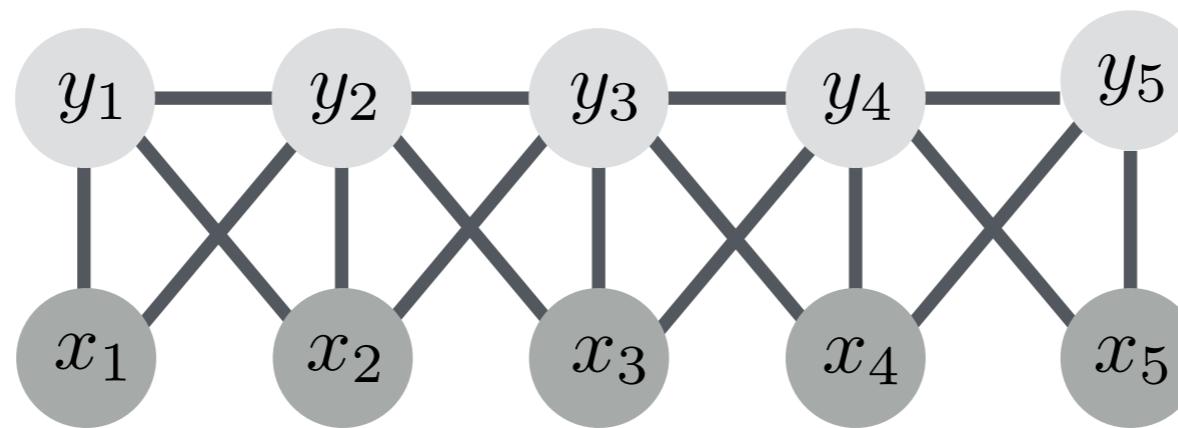
$$p(\mathbf{y}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})} \prod_f \Psi_f(\mathbf{y}, \mathbf{X})$$

$$\Psi_f(\mathbf{y}, \mathbf{X}) = \begin{cases} \psi_f(y_s, \mathbf{x}_{s-1}) &= \exp(a_{y_s}(\mathbf{x}_{s-1})) \\ \psi_f(y_s, \mathbf{x}_s) &= \exp(a_{y_s}(\mathbf{x}_s)) \\ \psi_f(y_s, \mathbf{x}_{s+1}) &= \exp(a_{y_s}(\mathbf{x}_{s+1})) \\ \psi_f(y_s, y_{s+1}) &= \exp(V_{y_s, y_{s+1}}) \end{cases}$$

Einschub: Conditional Random Field

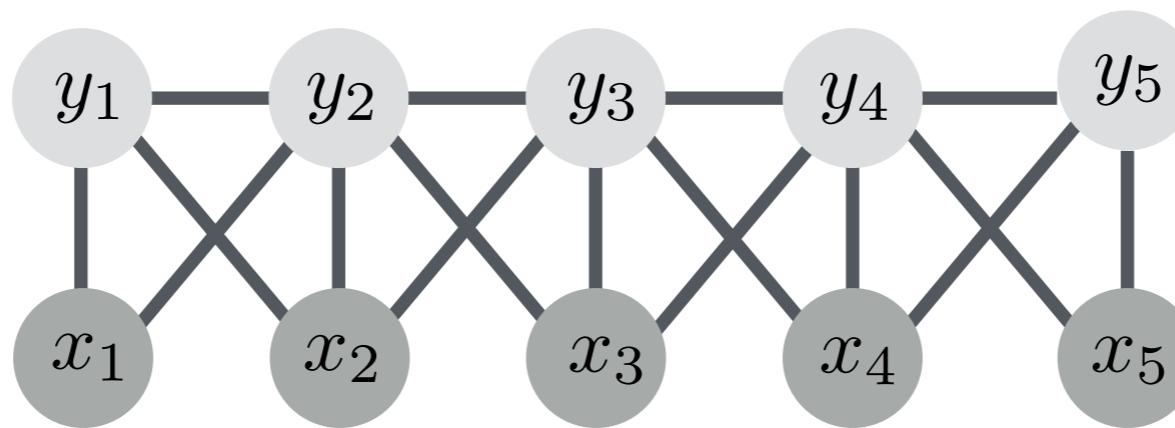
Visualisierung

hier: als Markov Netzwerk



Einschub: Conditional Random Field

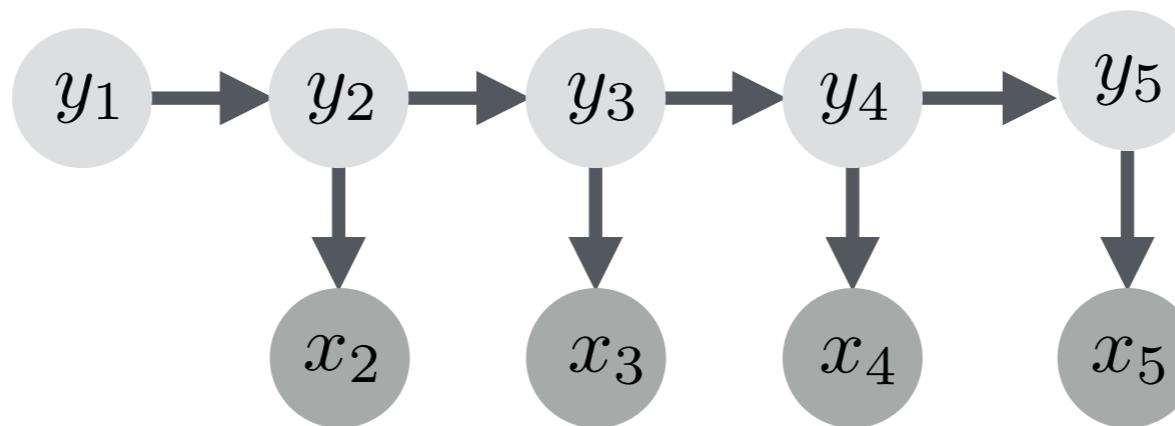
Ungerichteter Graph



Einschub: Conditional Random Field

Gerichteter Graph

hier: Hidden Markov Model

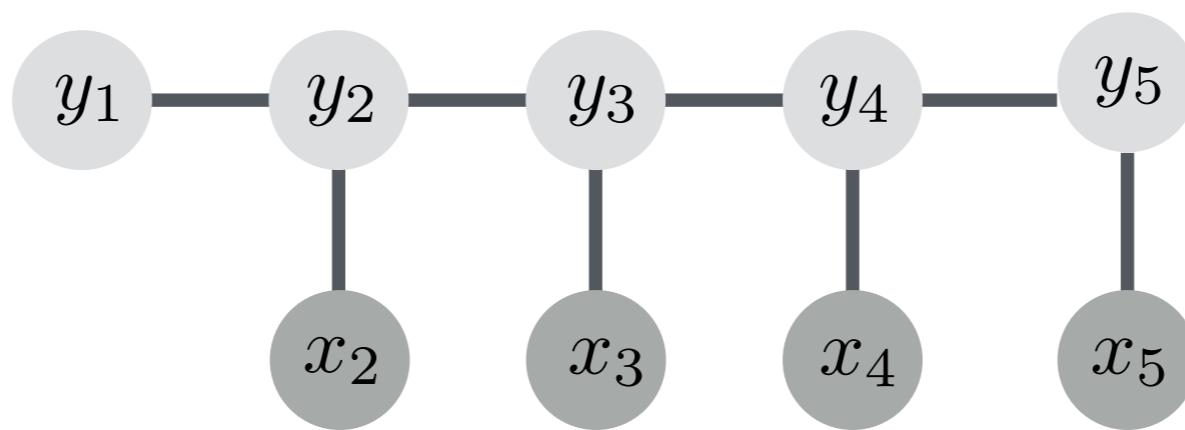


$$\psi_f(y_s, \mathbf{x}_s) = p(\mathbf{x}_s | y_s)$$

$$\psi_f(y_s, y_{s+1}) = p(y_{s+1} | y_s)$$

Einschub: Conditional Random Field

Ungerichteter Graph

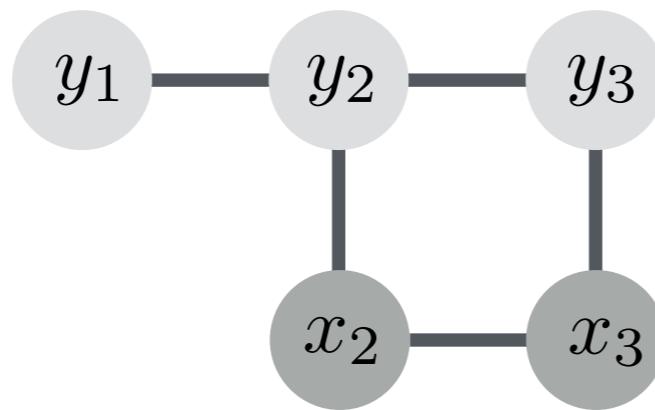


$$\psi_f(y_s, \mathbf{x}_s)$$

$$\psi_f(y_s, y_{s+1})$$

Einschub: Conditional Random Field

Ungerichteter Graph



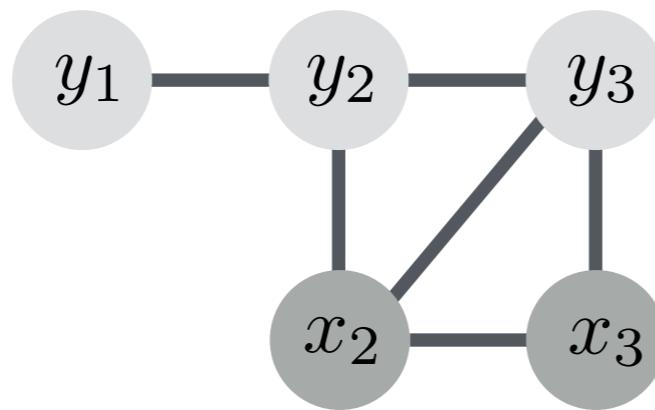
$$P(y_1, y_2, y_3, x_2, x_3) = \psi_f(y_1, y_2) \cdot \psi_f(y_2, y_3) \cdot \psi_f(y_3, x_3) \cdot \psi_f(y_2, x_2) \cdot \psi_f(x_2, x_3)$$

$$P(\mathbf{X}) = \frac{1}{Z} \prod_{c \in \text{cliques}(G)} \psi_f(x_c)$$

© B. Huang

Einschub: Conditional Random Field

Ungerichteter Graph



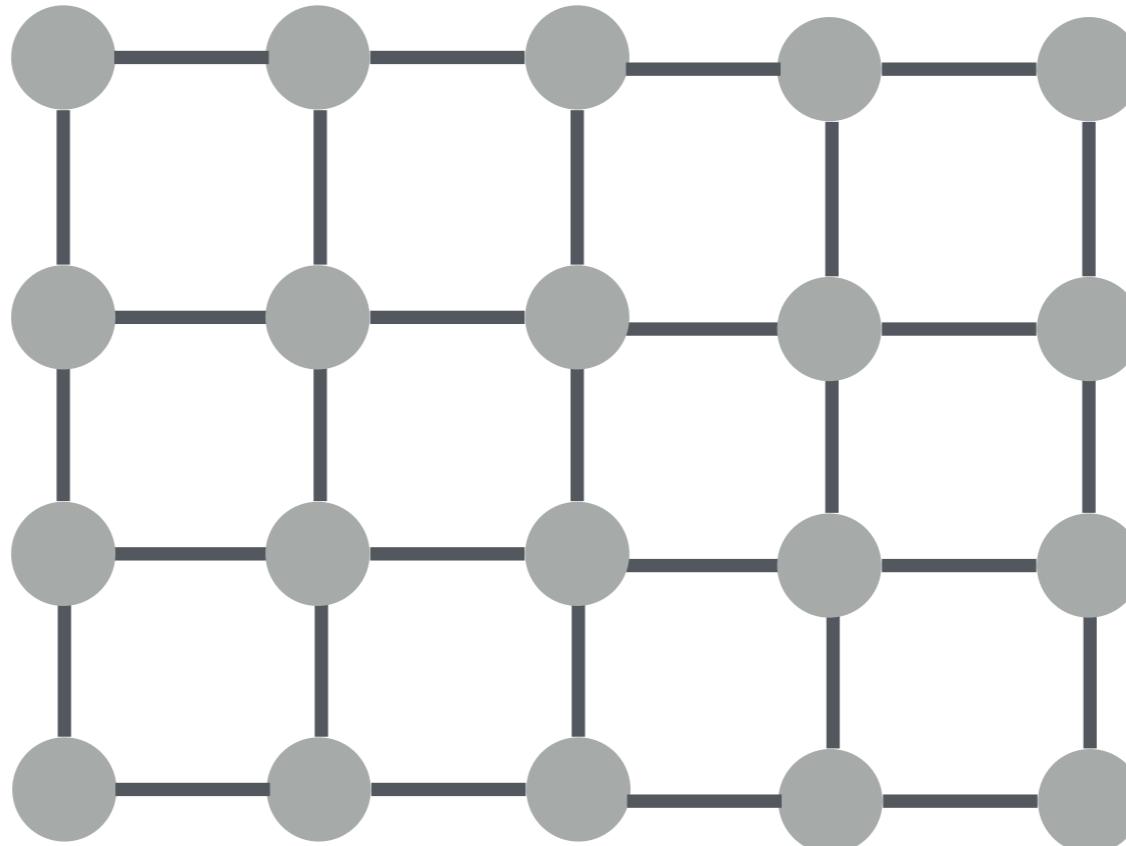
$$P(y_1, y_2, y_3, x_2, x_3) =$$

$$P(\mathbf{X}) = \frac{1}{Z} \prod_{c \in \text{cliques}(G)} \psi_f(x_c)$$

© B. Huang

Einschub: Conditional Random Field

Anwendung: Bildsegmentierung



$$E(\boldsymbol{x}) = \sum_i \psi_u(x_i) + \sum_{i,j, i \neq j} \psi_f(x_i, x_j)$$

© Krähenbühl et al., 2011

Einschub: Conditional Random Field

Anwendung: Bildsegmentierung

$$E(\mathbf{x}) = \sum_i \psi_u(x_i) + \sum_{i,j,i \neq j} \psi_f(x_i, x_j)$$

$$\psi_f(x_i, x_j) = \nu(x_i, x_j) k(f_i, f_j)$$

Einschub: Conditional Random Field

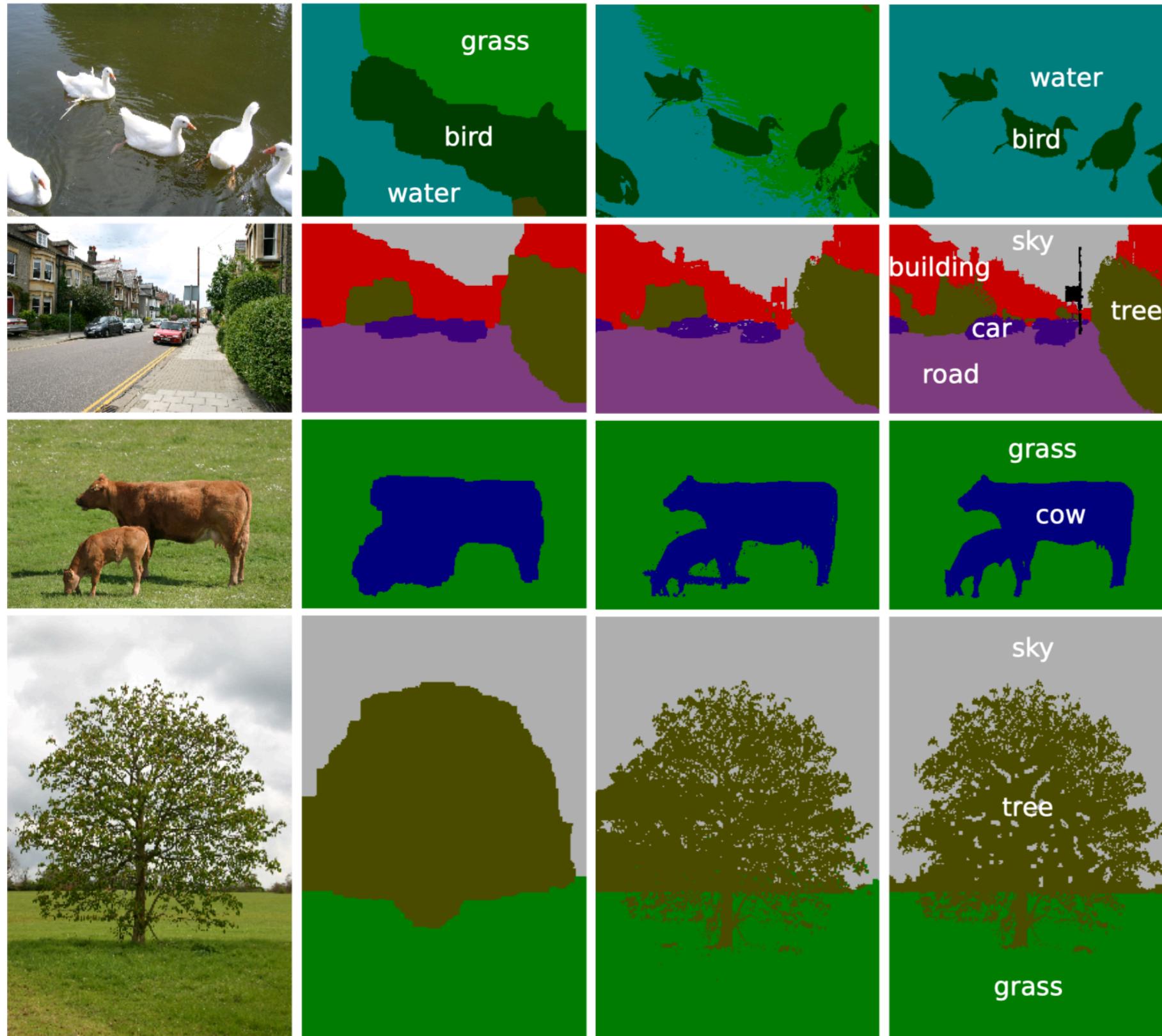
Anwendung: Bildsegmentierung

$$E(\mathbf{x}) = \sum_i \psi_u(x_i) + \sum_{i,j,i \neq j} \psi_f(x_i, x_j)$$

$$\psi_f(x_i, x_j) = \nu(x_i, x_j) k(f_i, f_j)$$

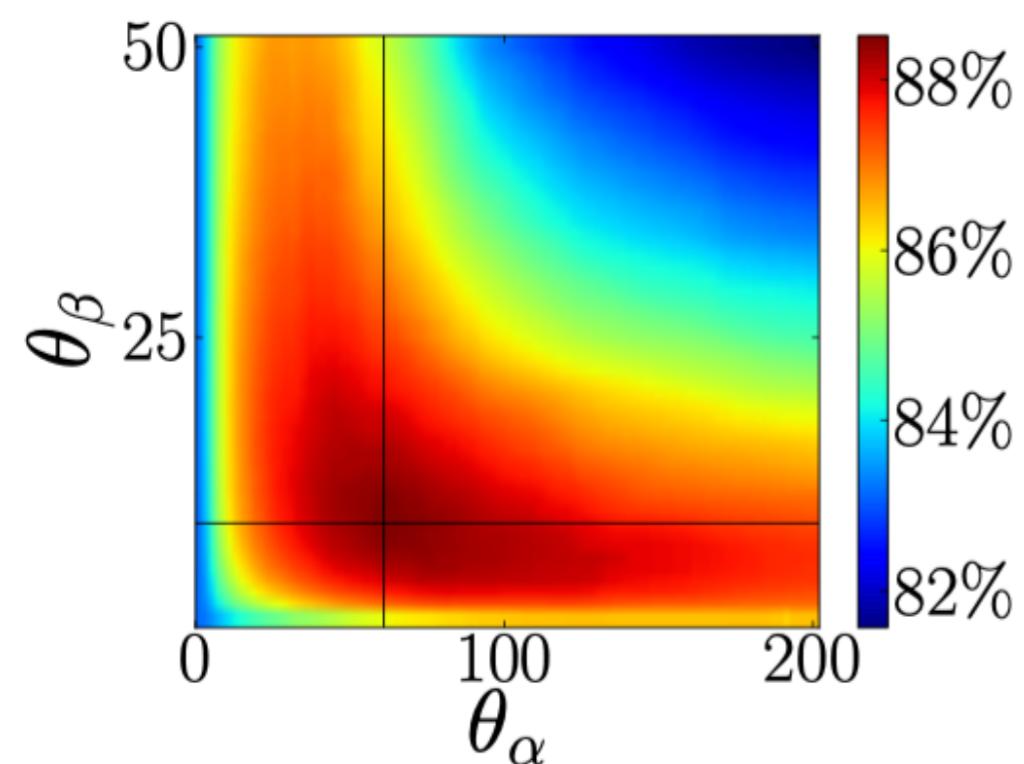
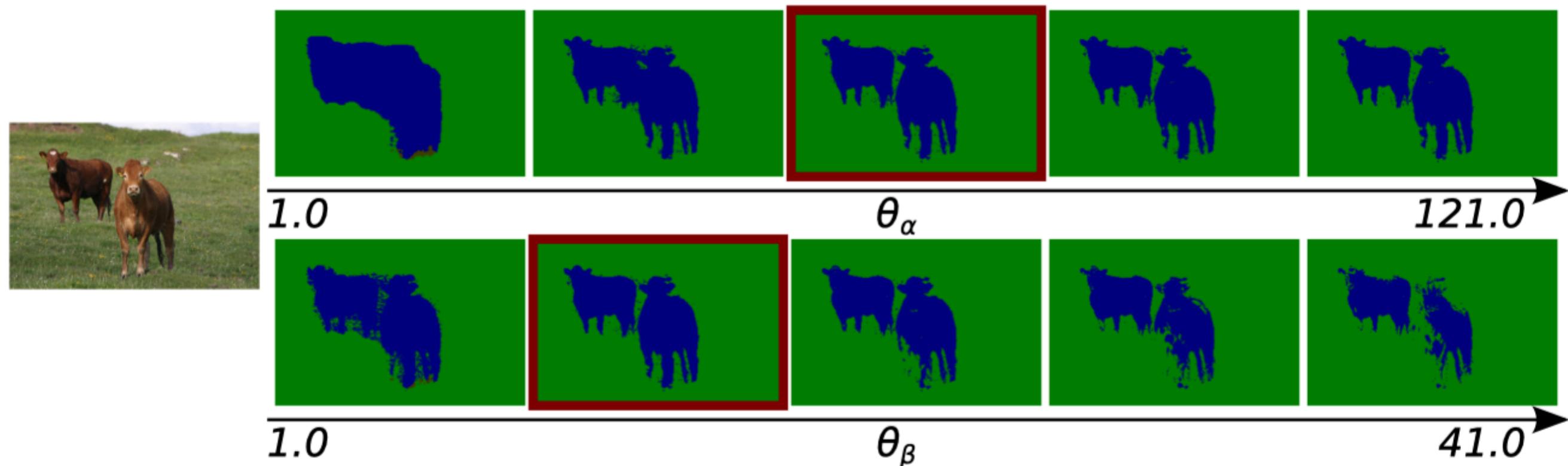
$$k(f_i, f_j) = w_1 \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right) + w_2 \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right)$$

Einschub: Conditional Random Field



© Krähenbühl et al., 2011

Einschub: Conditional Random Field

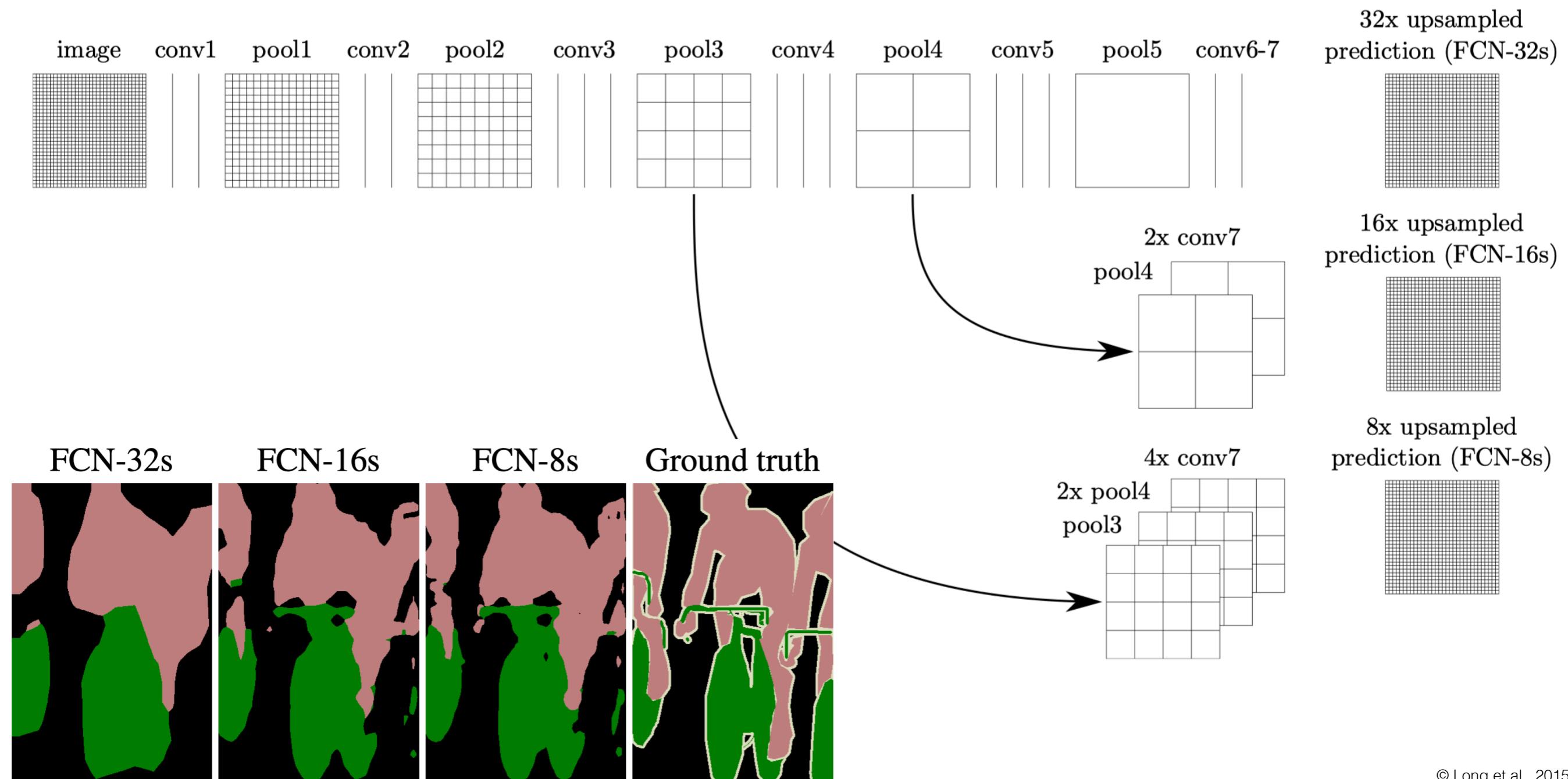


$$k(\mathbf{f}_i, \mathbf{f}_j) = w_1 \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right) + w_2 \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right)$$

Semantische Segmentierung

DeepLab v1

folgt der Idee von Long et al.

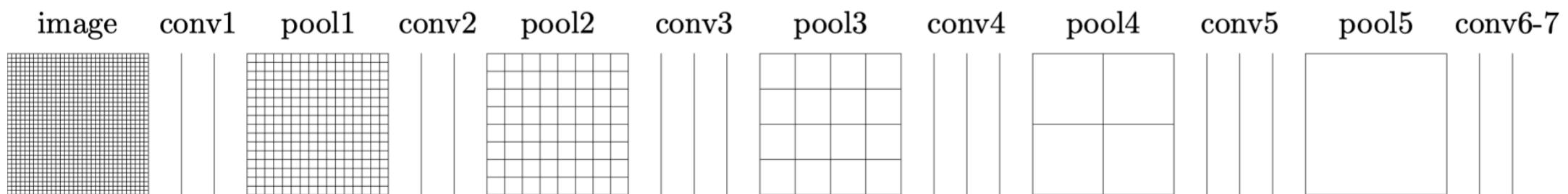


© Long et al., 2015

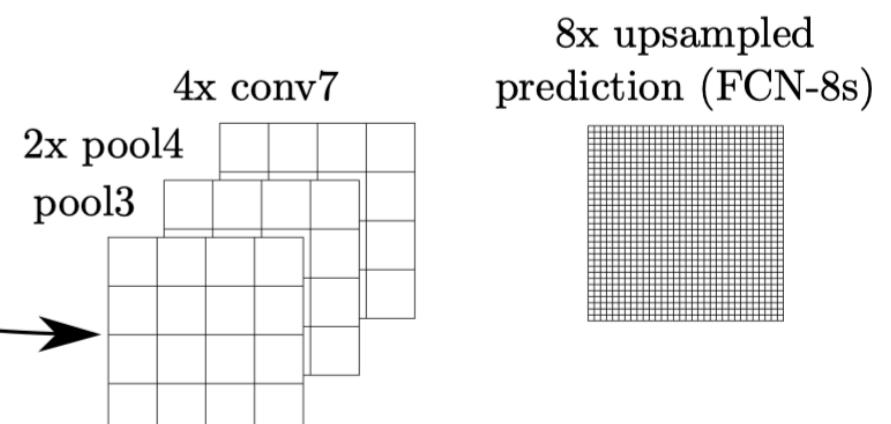
Semantische Segmentierung

DeepLab v1

folgt der Idee von Long et al.



Änderungen:

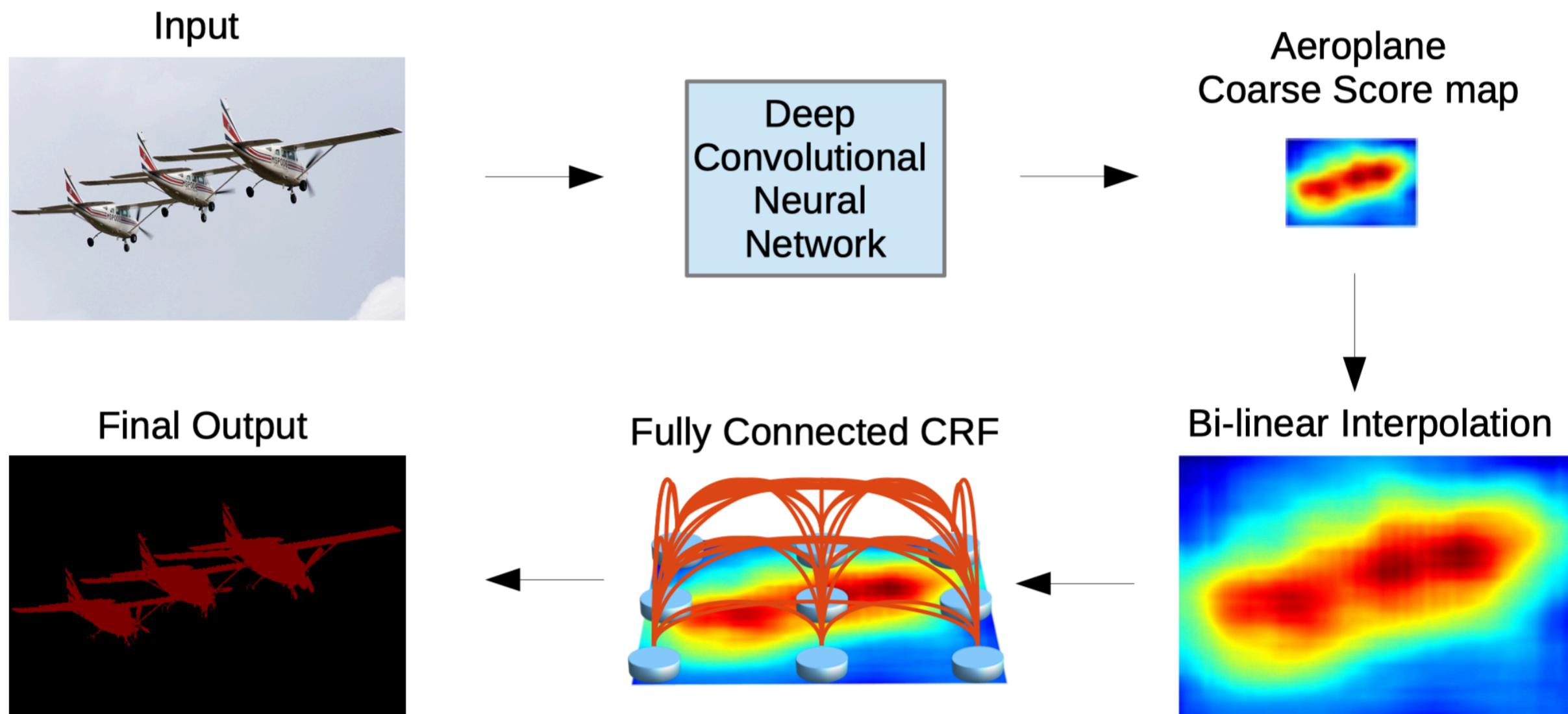


© Long et al., 2015

Semantische Segmentierung

DeepLab v1

nutzt Conditional Random Fields (CRF)

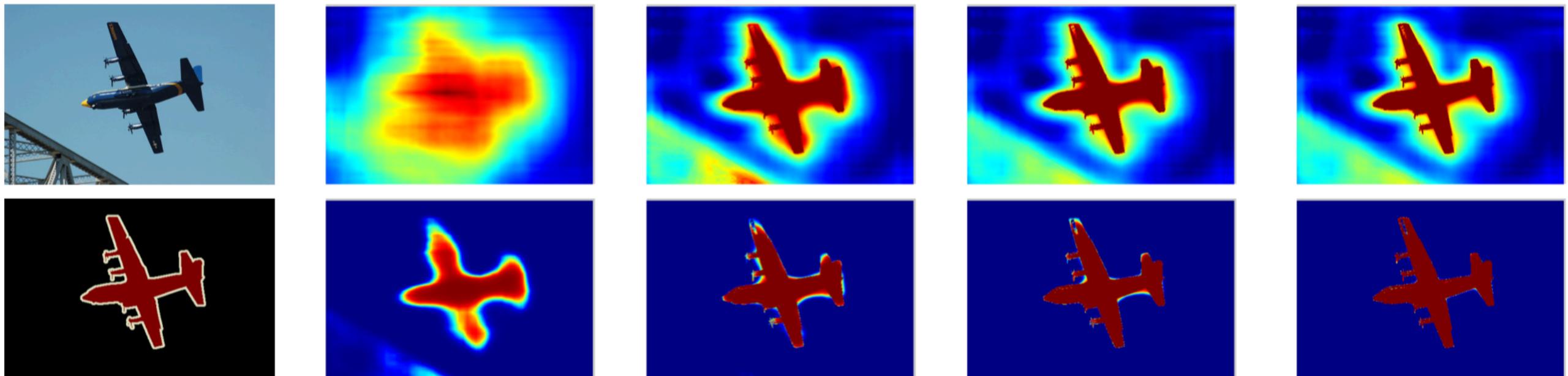


© Liang-Chieh Chen et al., 2016

Semantische Segmentierung

DeepLab v1

nutzt Conditional Random Fields (CRF)



Semantische Segmentierung

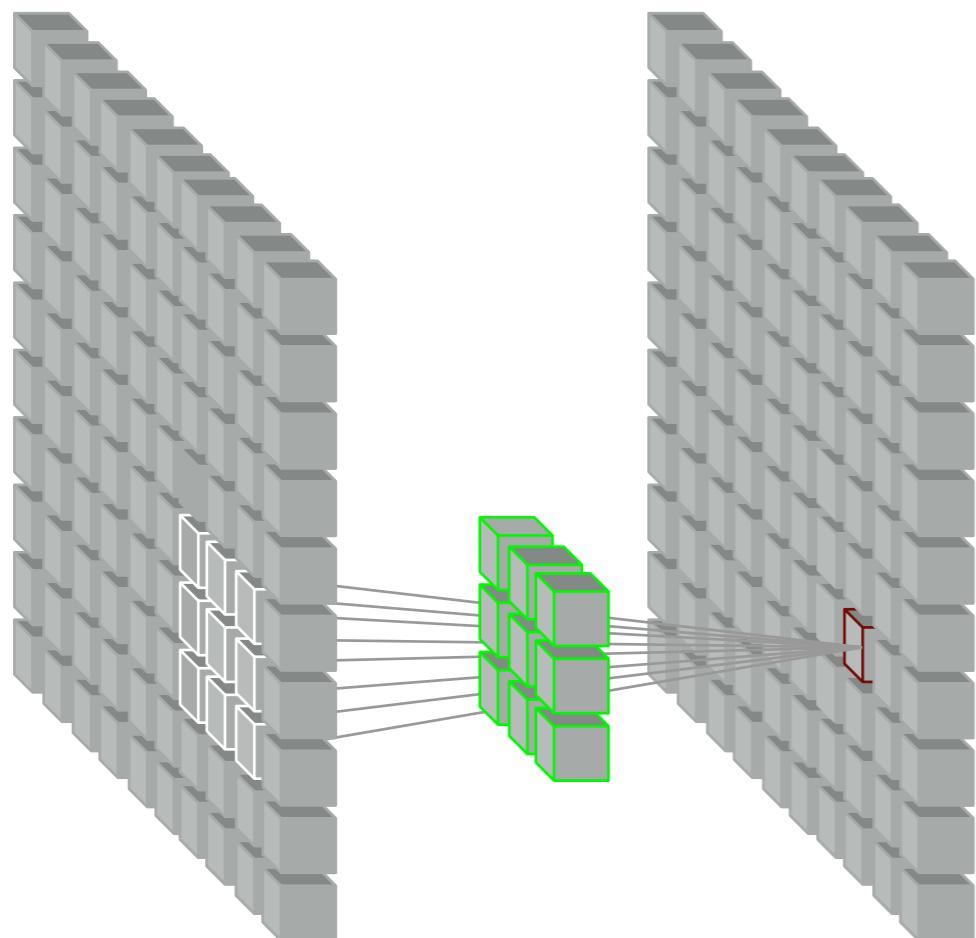
DeepLab v1



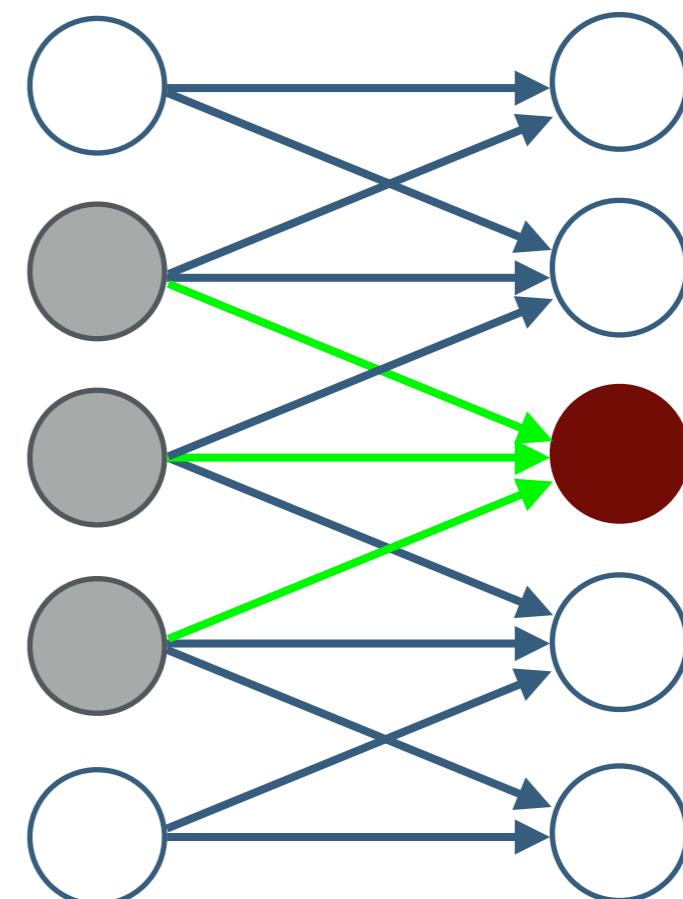
© Liang-Chieh Chen et al., 2016

Semantische Segmentierung

Faltung bisher



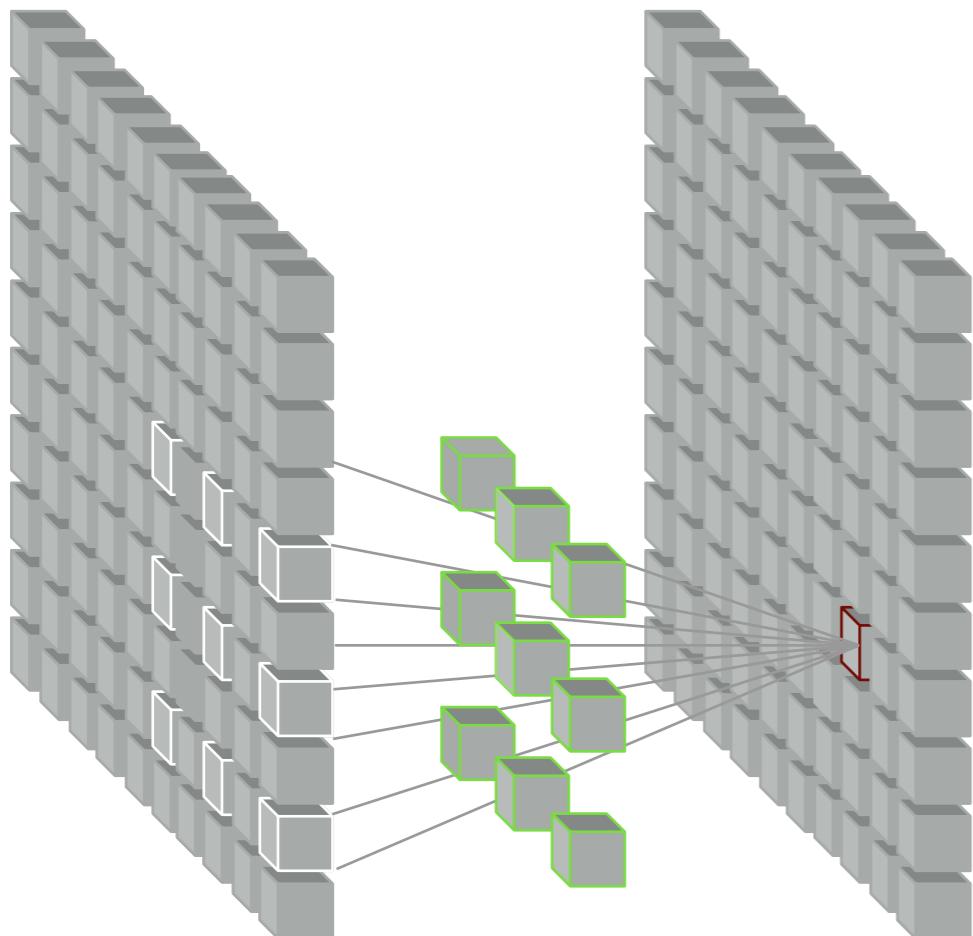
$$g(p, q) = \sum_{i=-\lfloor \frac{I}{2} \rfloor}^{\frac{I}{2}} \sum_{j=-\lfloor \frac{J}{2} \rfloor}^{\frac{J}{2}} f(p+i, q+j)h(i, j)$$



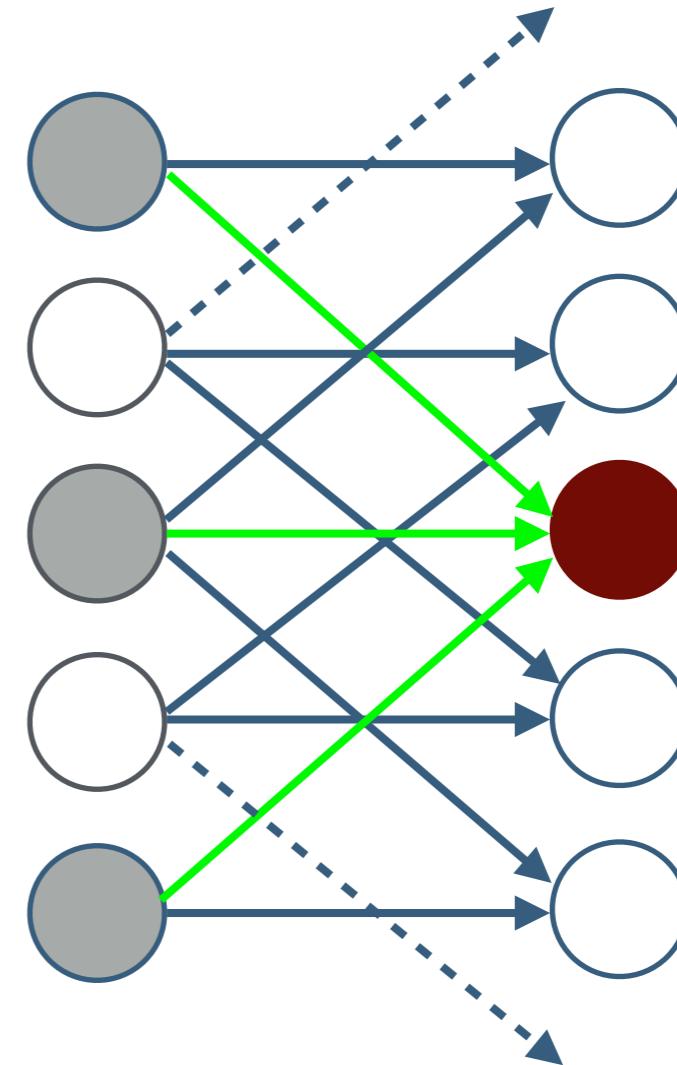
Semantische Segmentierung

Faltung mit Input Stride

>> Atrous Convolution



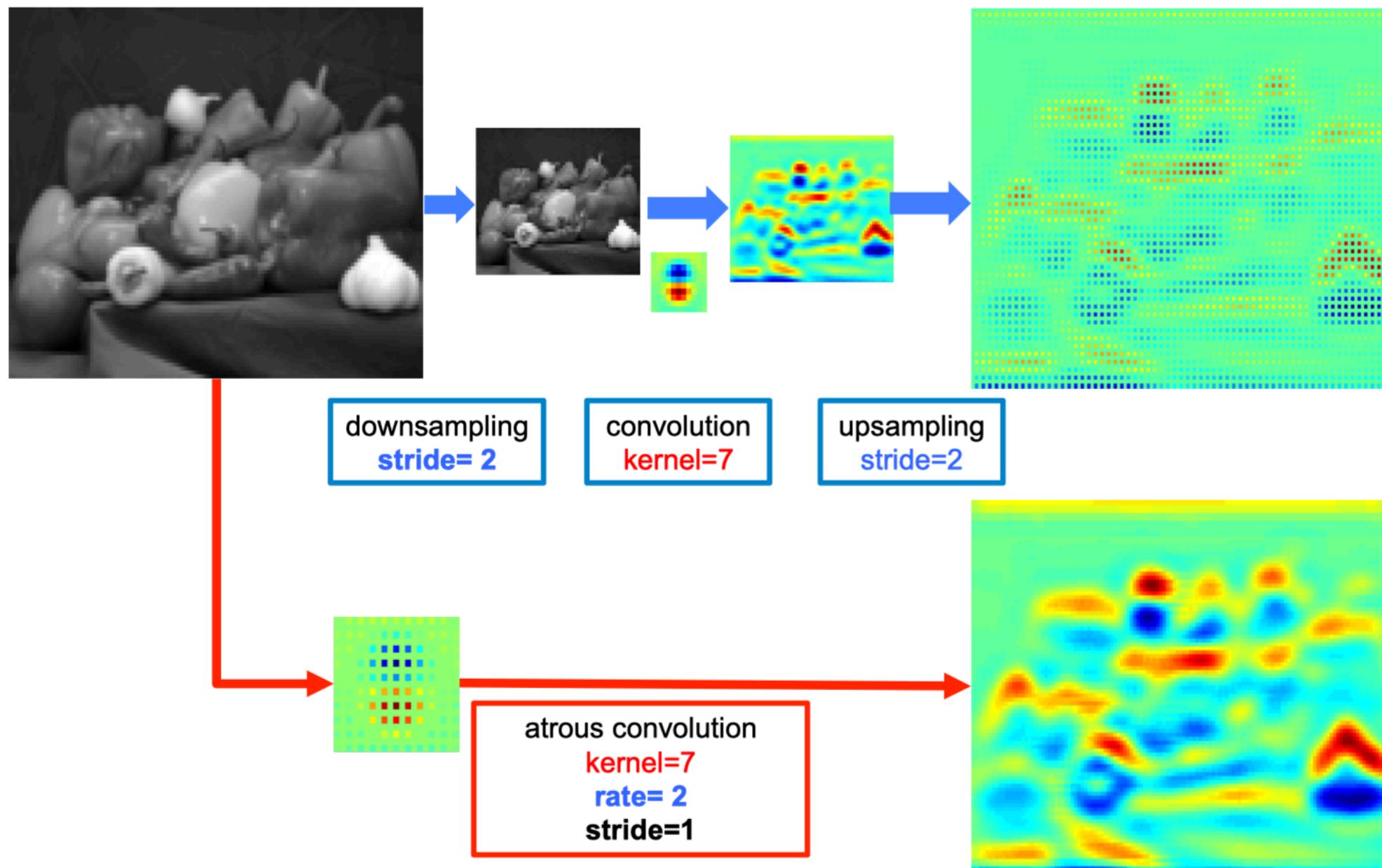
$$g(p, q) = \sum_{i=-\lfloor \frac{I}{2} \rfloor}^{\frac{I}{2}} \sum_{j=-\lfloor \frac{J}{2} \rfloor}^{\frac{J}{2}} f(p + r \cdot i, q + r \cdot j) h(i, j)$$



Semantische Segmentierung

Faltung mit Input Stride

>> Atrous Convolution

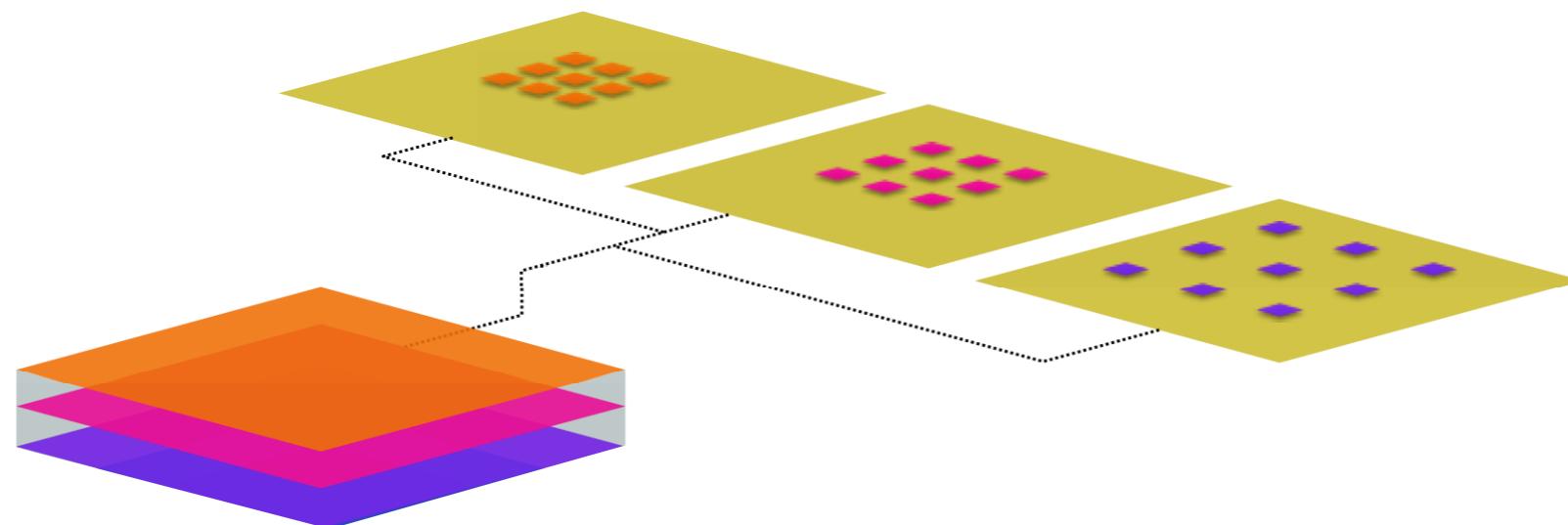


© Liang-Chieh Chen et al., 2017

Semantische Segmentierung

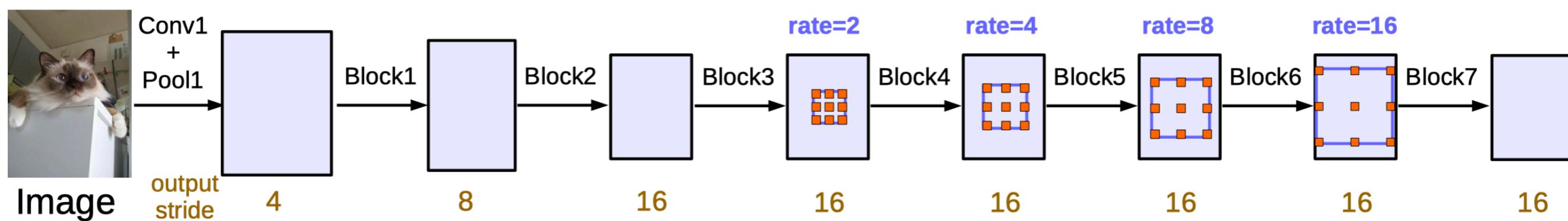
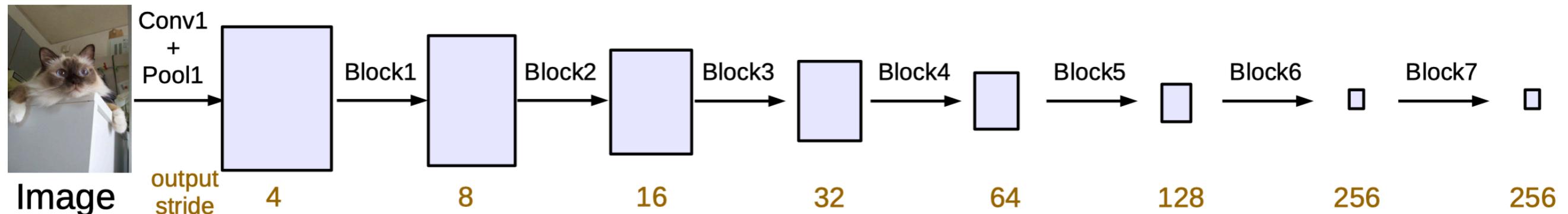
DeepLab v2

Unterschied zu DeepLab v1: Atrous Spatial Pyramid Pooling



Semantische Segmentierung

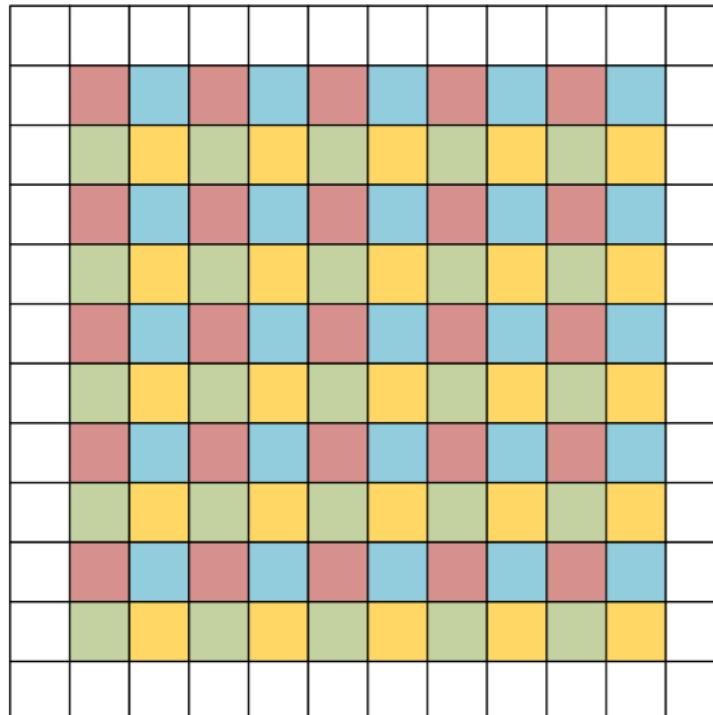
Varianten tiefer Faltungsnetze



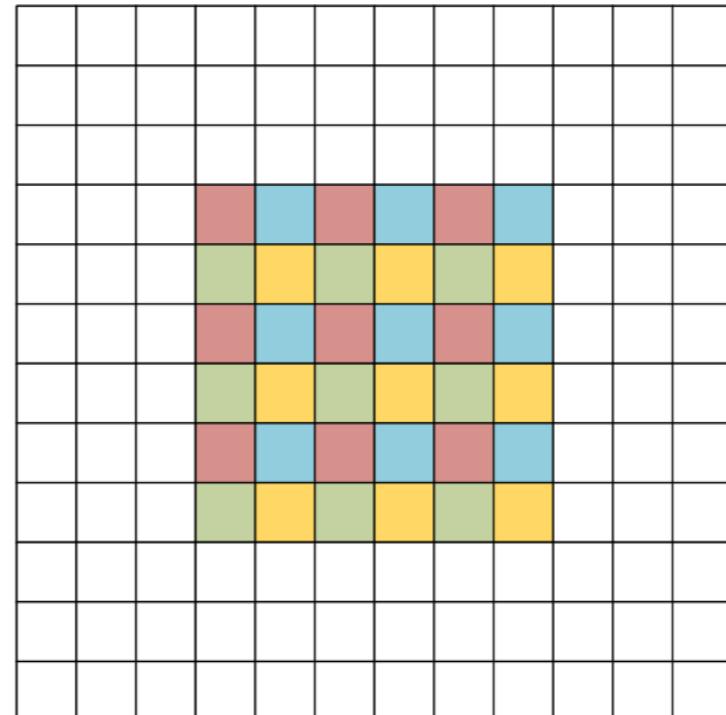
© Liang-Chieh Chen et al., 2017

Semantische Segmentierung

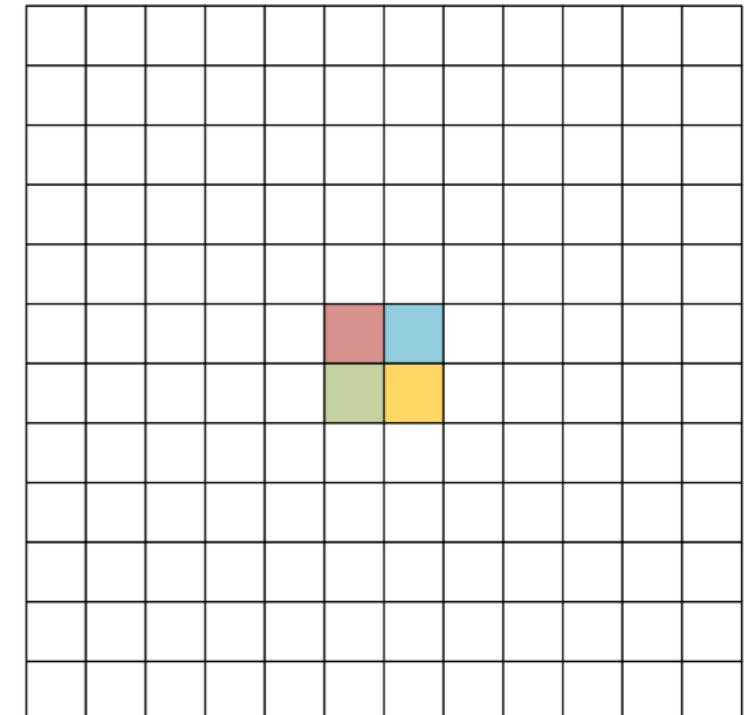
Multigrid



Layer $i-2$



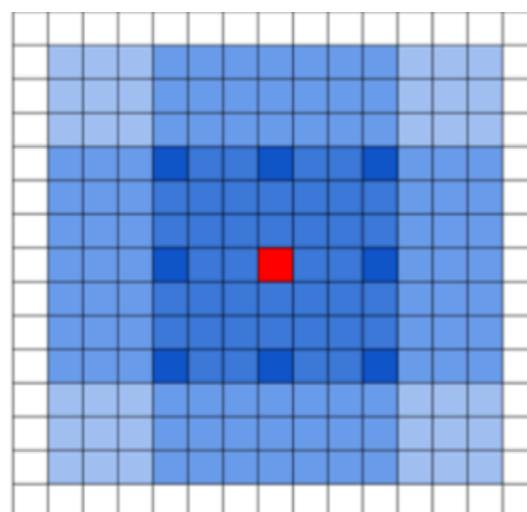
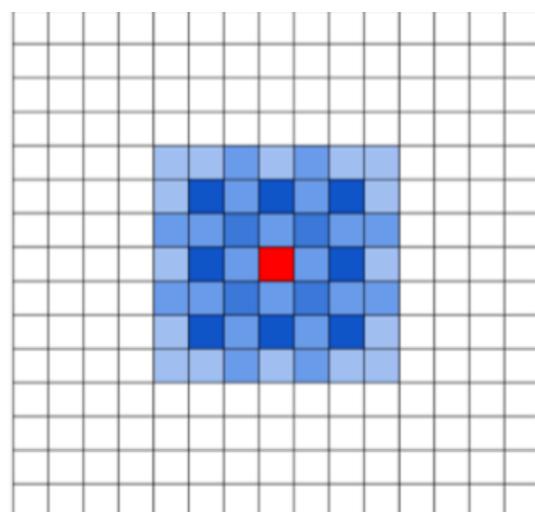
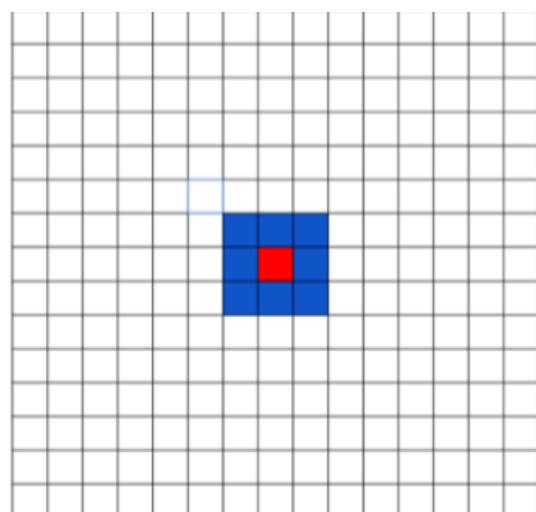
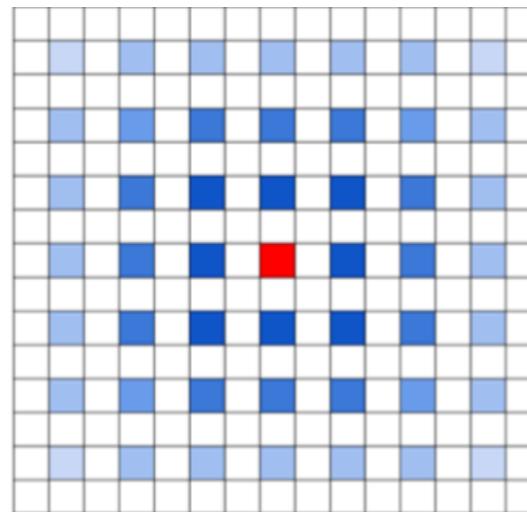
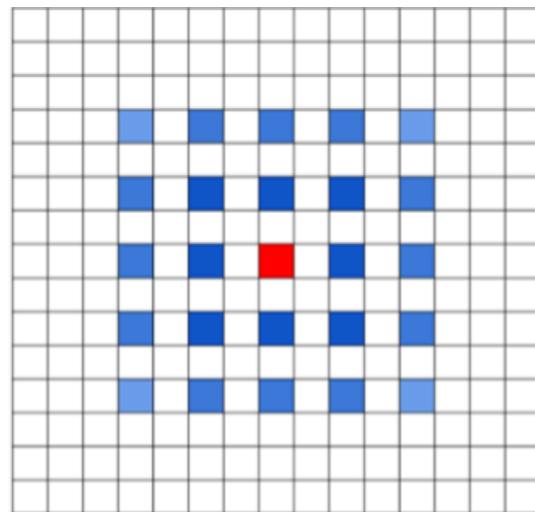
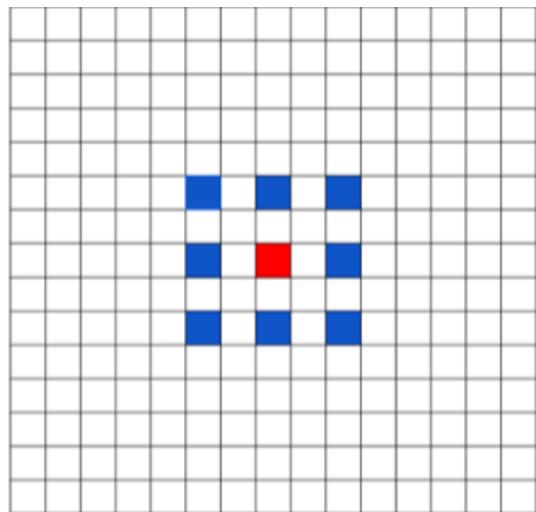
Layer $i-1$



Layer i

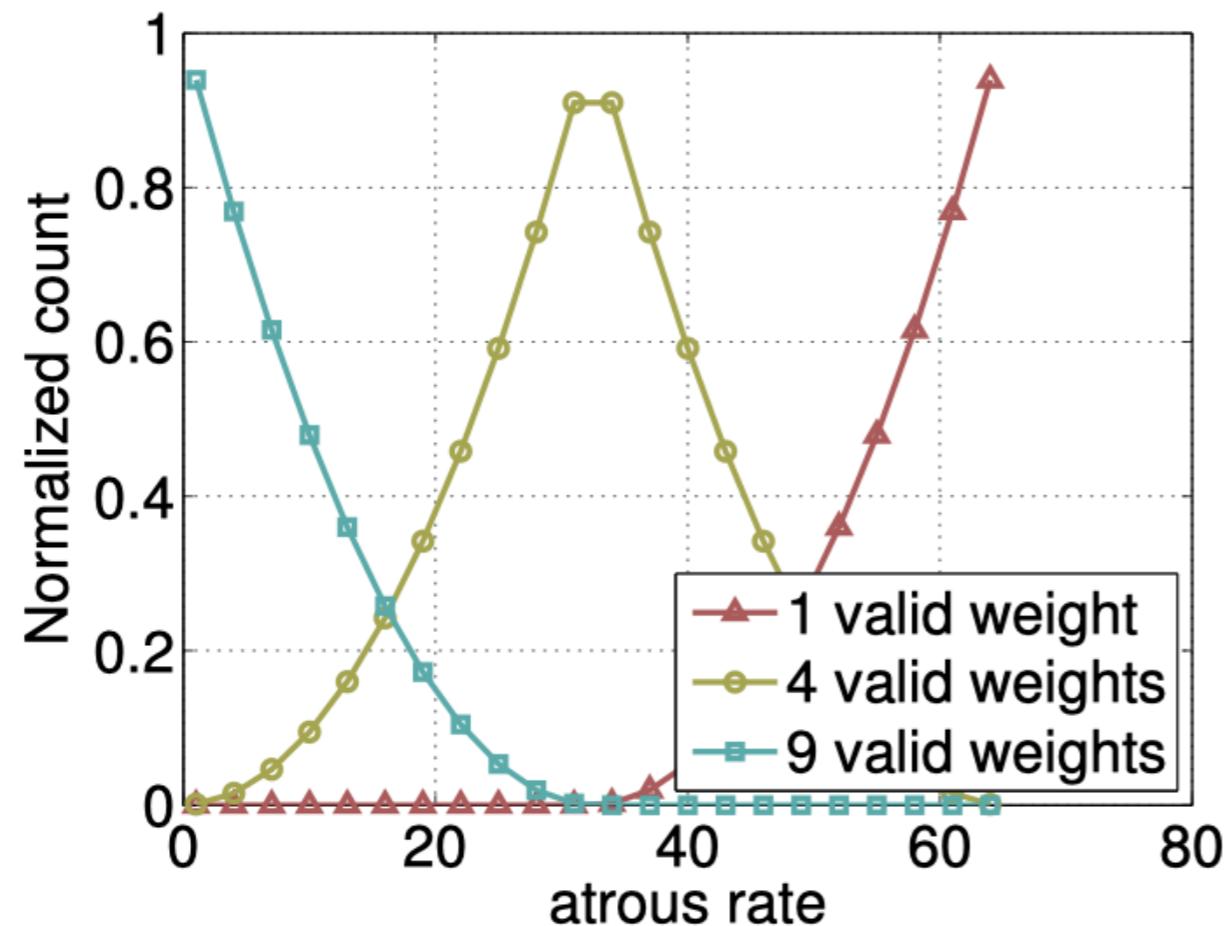
Semantische Segmentierung

Multigrid



Semantische Segmentierung

Faltungskern bei atrous convolution



Semantische Segmentierung

DeepLab v3

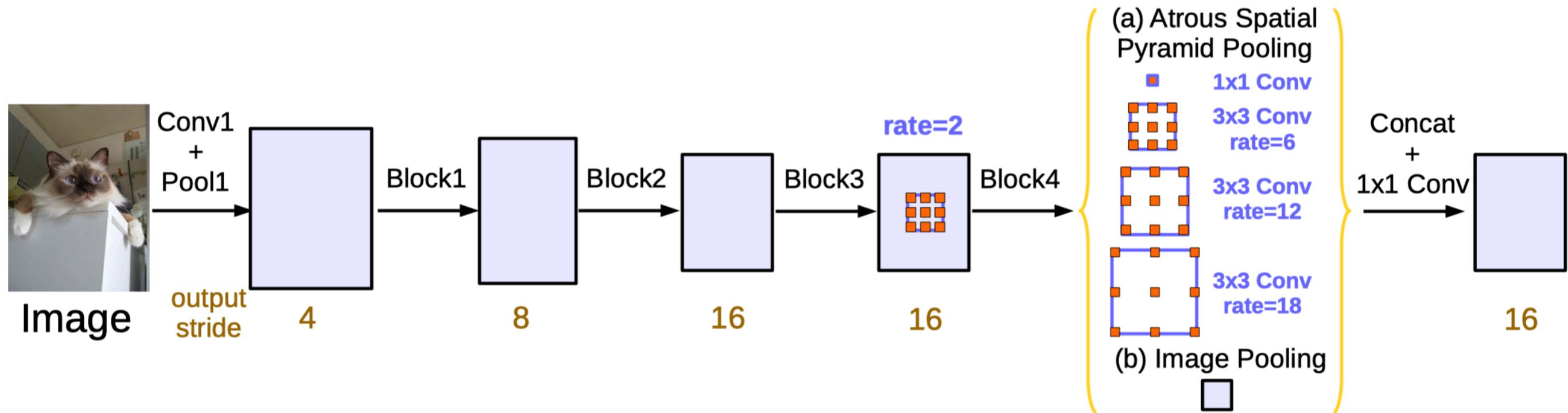
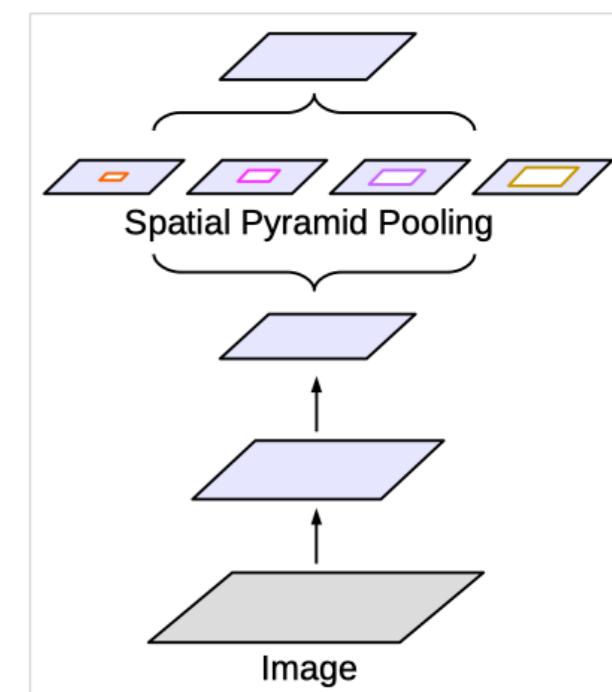
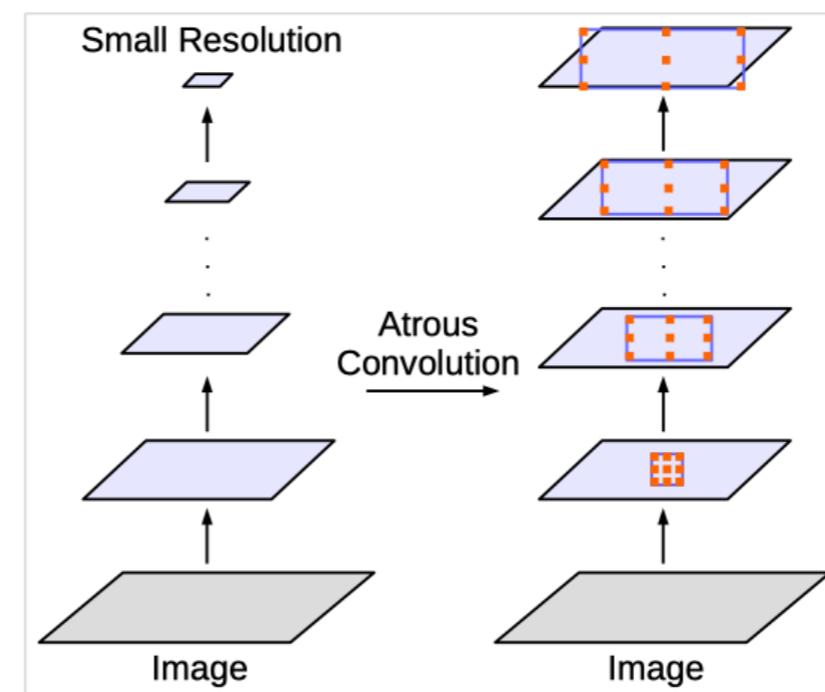
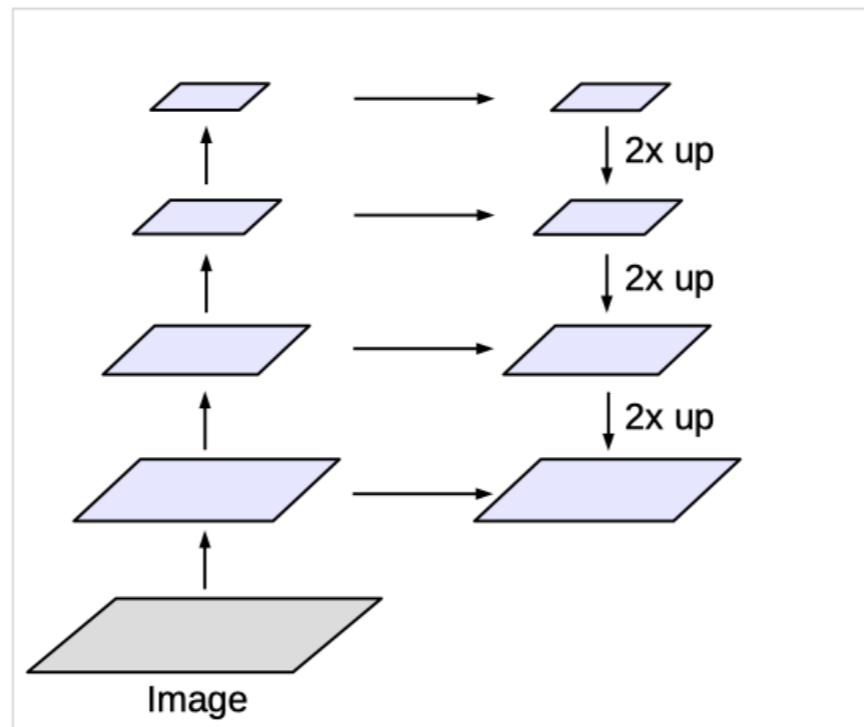
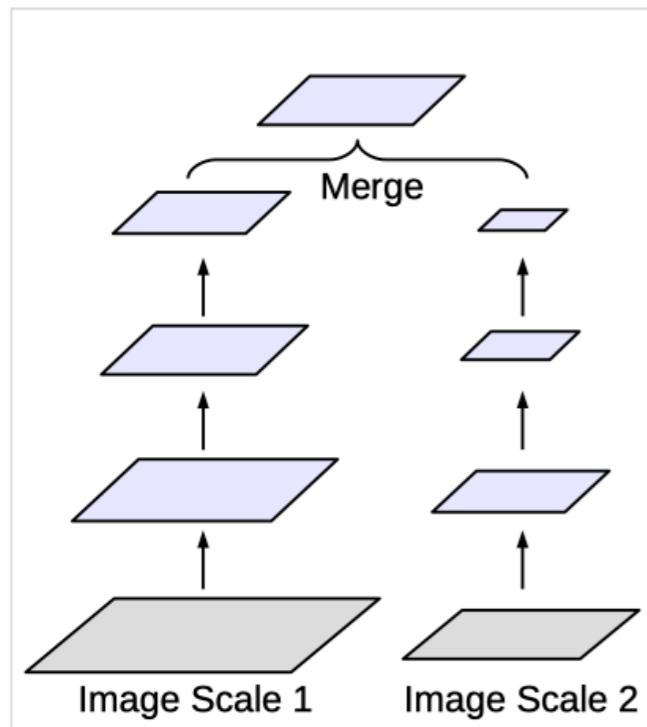


Image Pooling:

Semantische Segmentierung

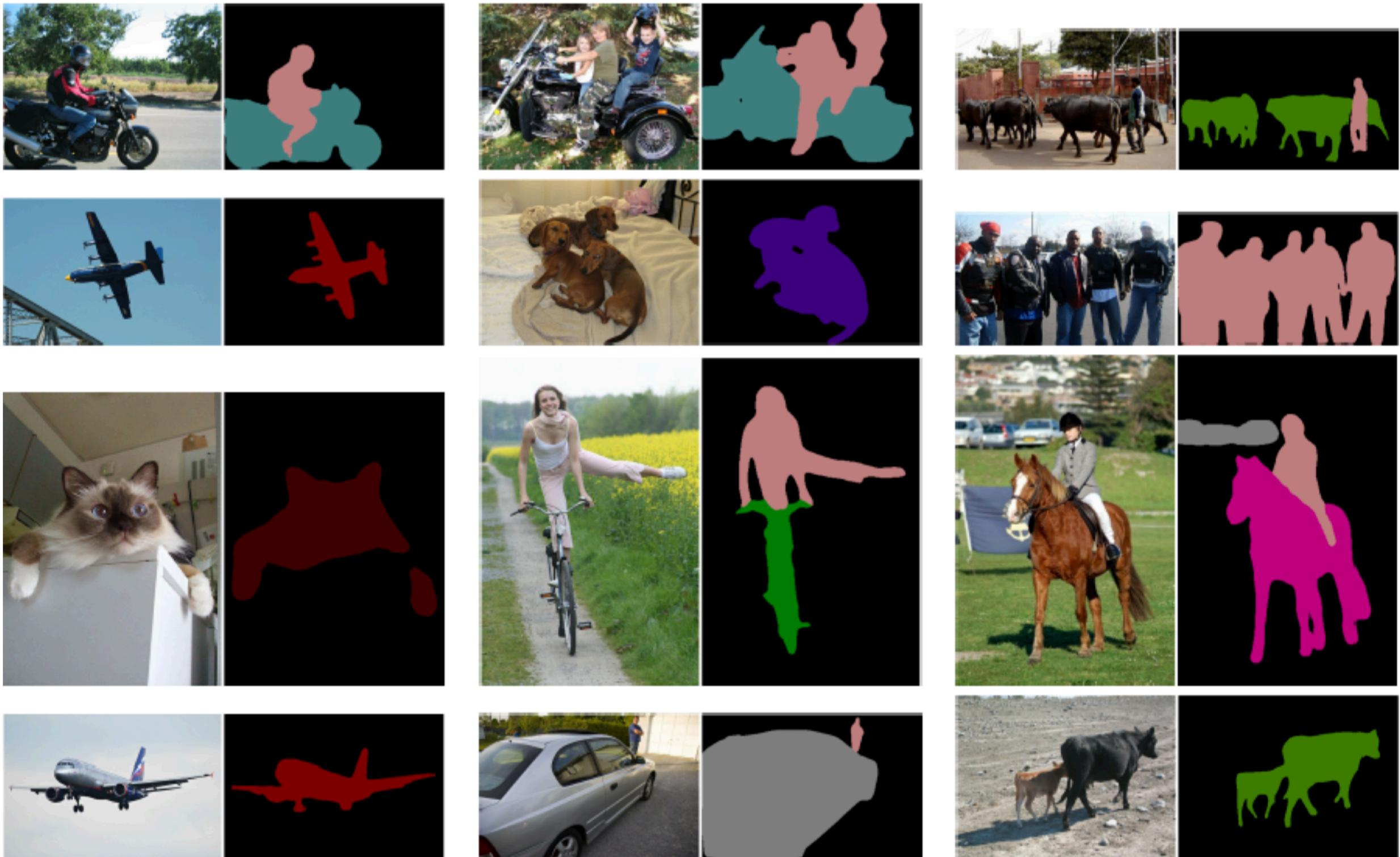
Kontext auf verschiedenen Skalen



© Liang-Chieh Chen et al., 2017

Semantische Segmentierung

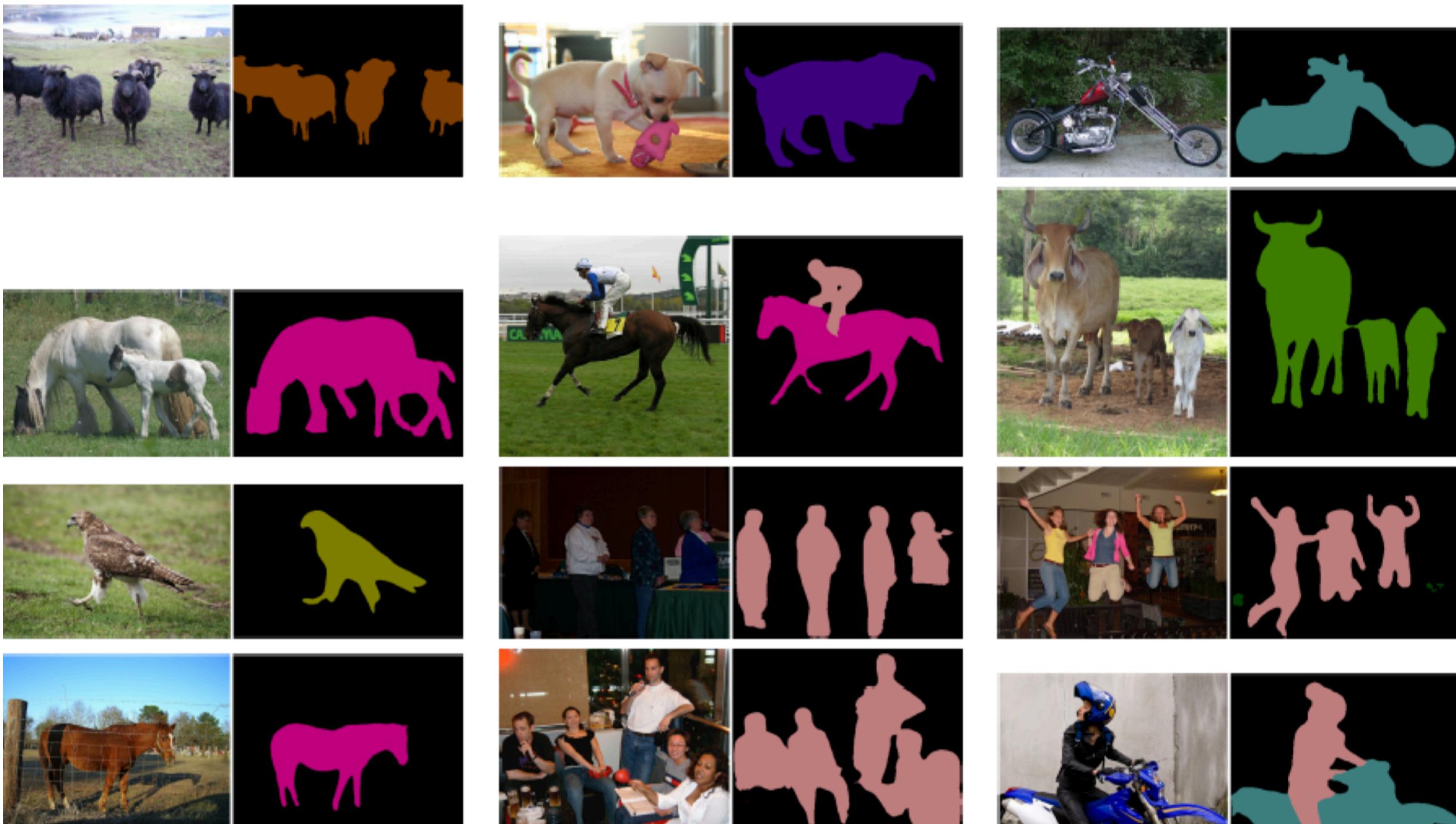
DeepLab v3



© Liang-Chieh Chen et al., 2017

Semantische Segmentierung

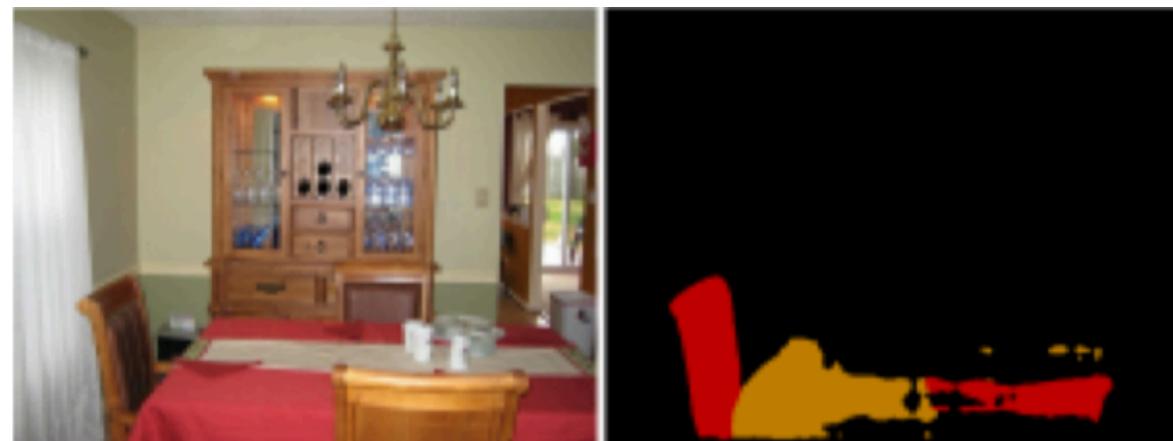
DeepLab v3



© Liang-Chieh Chen et al., 2017

Semantische Segmentierung

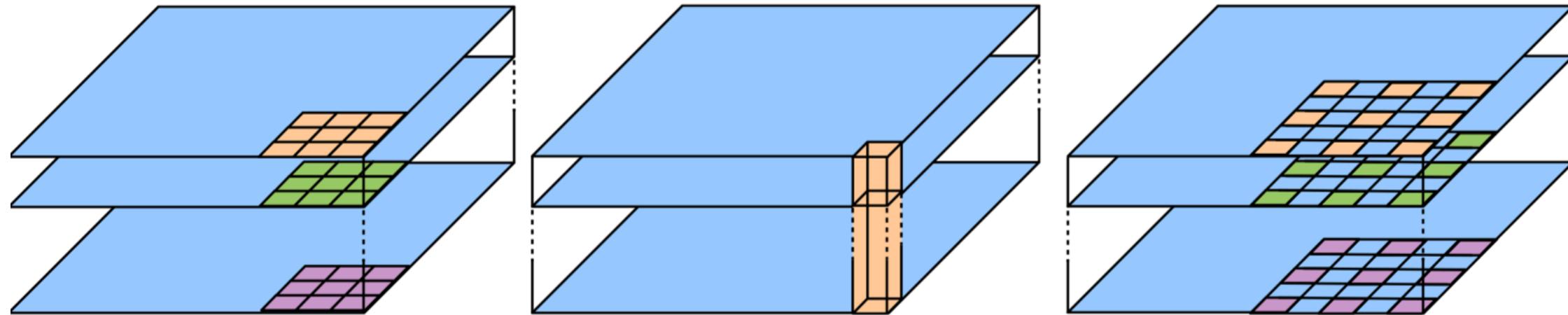
DeepLab v3



© Liang-Chieh Chen et al., 2017

Semantische Segmentierung

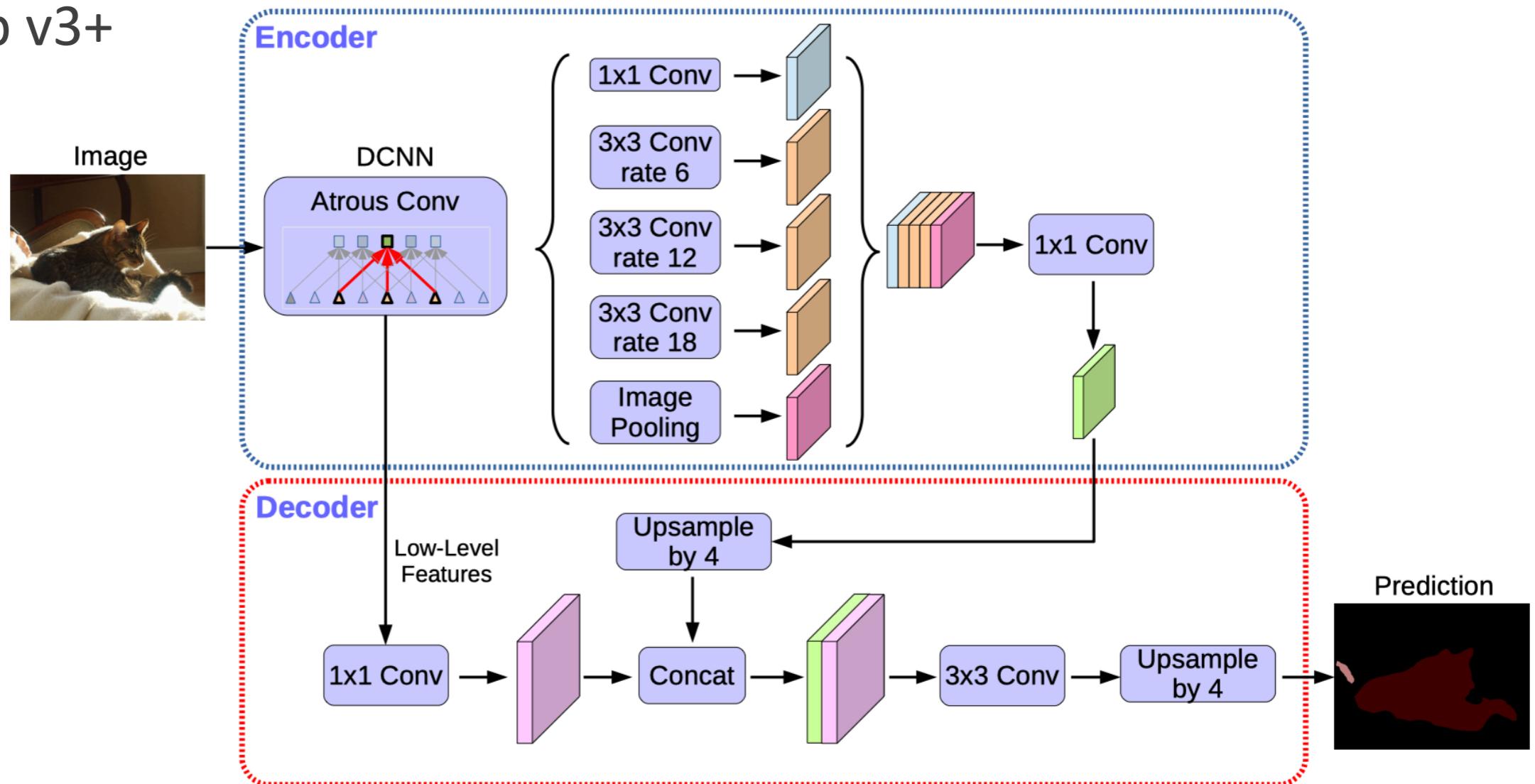
DeepLab v3+



© Liang-Chieh Chen et al., 2018

Semantische Segmentierung

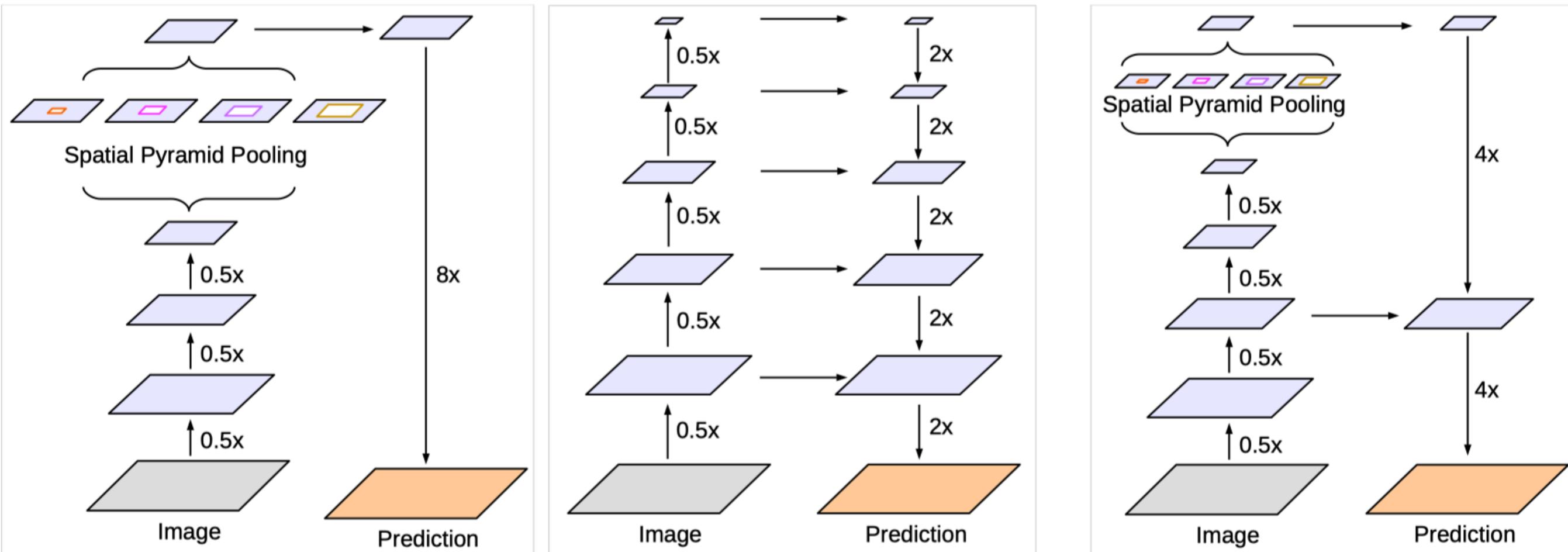
DeepLab v3+



© Liang-Chieh Chen et al., 2018

Semantische Segmentierung

DeepLab v3+



© Liang-Chieh Chen et al., 2018

Semantische Segmentierung

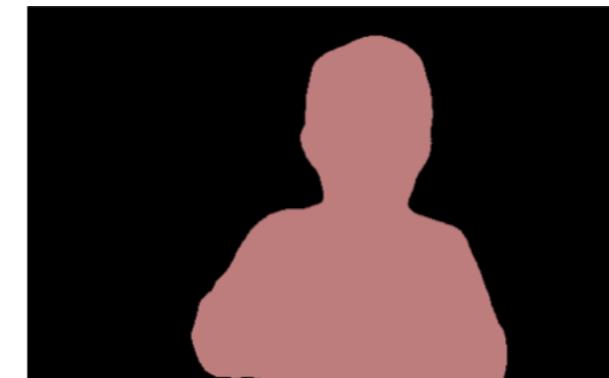
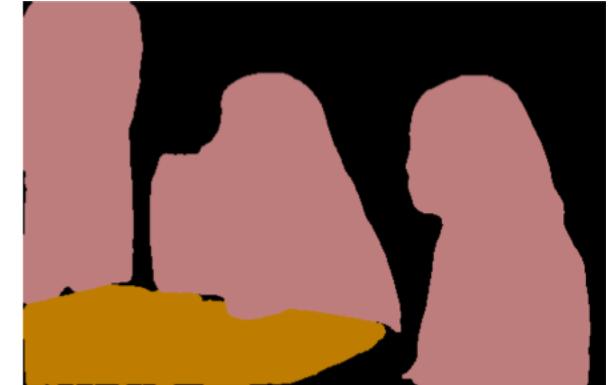
DeepLab v3+



© Liang-Chieh Chen et al., 2018

Semantische Segmentierung

DeepLab v3+



© Liang-Chieh Chen et al., 2018

Semantische Segmentierung

DeepLab v3+

