

COMPUTATIONAL INTELLIGENCE

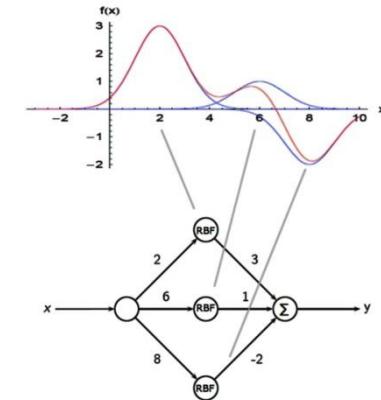
6. Deep Learning cont., Rekurrente Netze

Prof. Dr. Sven Behnke

Letzte Vorlesung

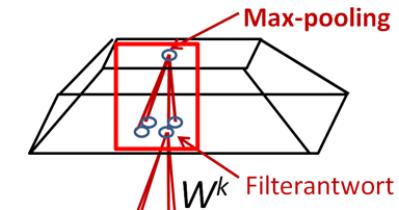
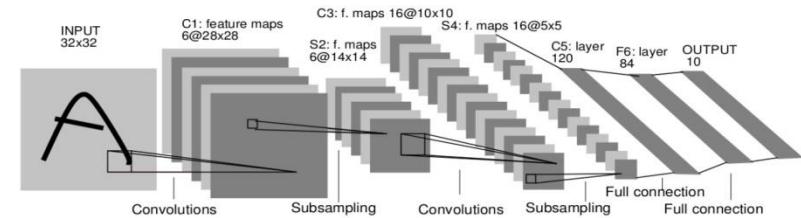
■ Radiale Basis-Funktions-Netzwerke

- Integrationsfunktion: Abstand $||x-c||$
- RBF-Aktivierungsfunktion: monoton fallend, z.B. Gaussglocke
=> lokale Effekte
- Initialisierung der Zentren unüberwacht
- Überwachtes Training der Gewichte der Ausgabeschicht



■ Konvolutionale Neuronale Netze

- Lokale Verbindungsstruktur
- Sharing von Parametern in Konvolution
- Bildartige Repräsentationen (Merkmalskarten)
- Pooling verringert Ortsauflösung
- Lernen mehrschichtiger Repräsentationen (Deep Learning)



Wettbewerb: ImageNet Challenge

- 1.2 Millionen Bilder
- 1000 Kategorien, kein Überlapp
- Teilmenge von 11 Millionen Bildern aus 15.000+ Kategorien
- Hierarchische Kategorienstruktur (WordNet)



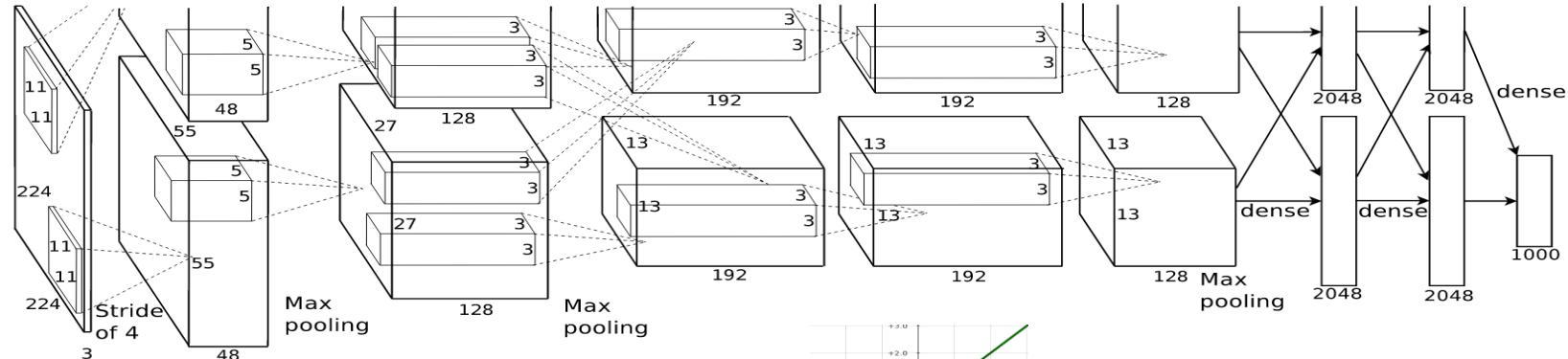
Golf cart (motor vehicle, self-propelled vehicle, wheeled vehicle, ...)



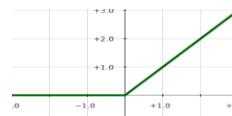
Egyptian cat (domestic cat, domestic animal, animal)

- Aufgabe: Erkennung der Objektkategorie
- Zusätzliche Detektionen werden nur gering bestraft
- Hierarchische Fehlerberechnung

AlexNet: Großes Konvolutionales Netzwerk



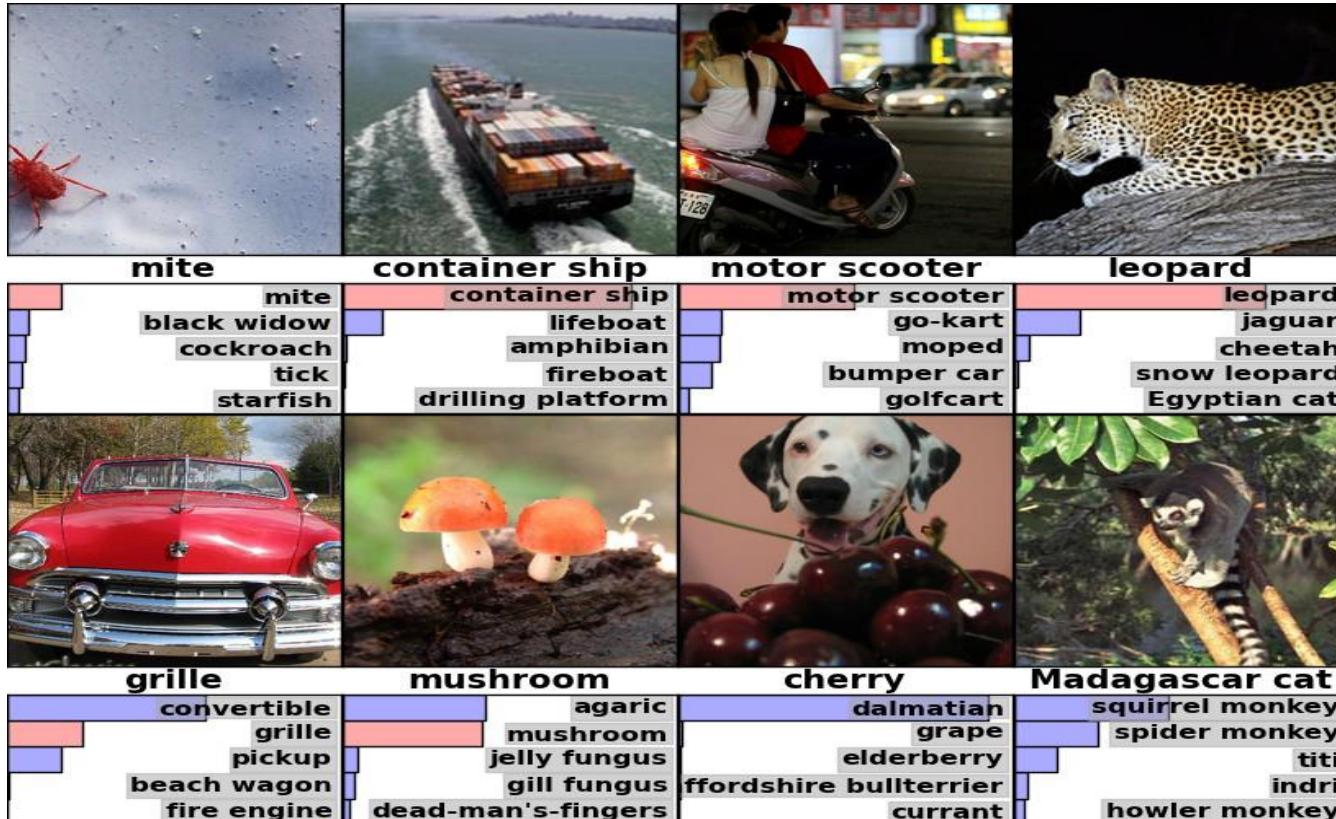
- Gleichrichtende Transferfunktionen
- 650,000 Neuronen
- 60,000,000 Parameter
- 630,000,000 Verbindungen
- Trainiert mit Dropout und
Datenaugmentierung
- Testen von 10 Teilbildern
- ILSVRC-2012: Top-5-Fehler 15.3%



96 gelernte Filter in der ersten Schicht

[Krizhevsky et al. NIPS 2012]

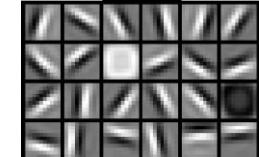
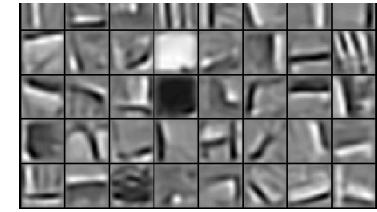
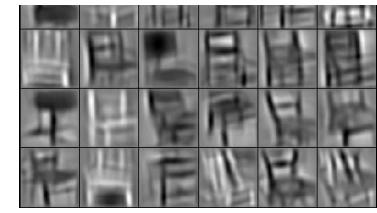
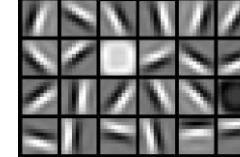
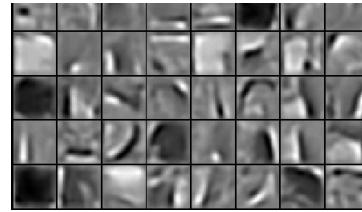
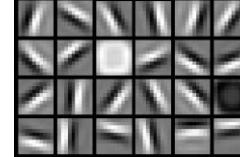
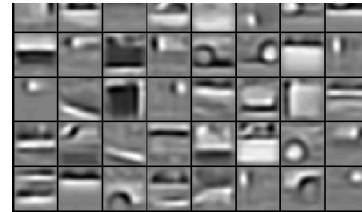
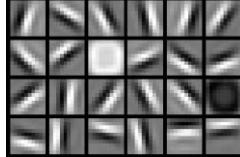
Klassifikation der Validierungsmenge



[Krizhevsky et al.
NIPS 2012]

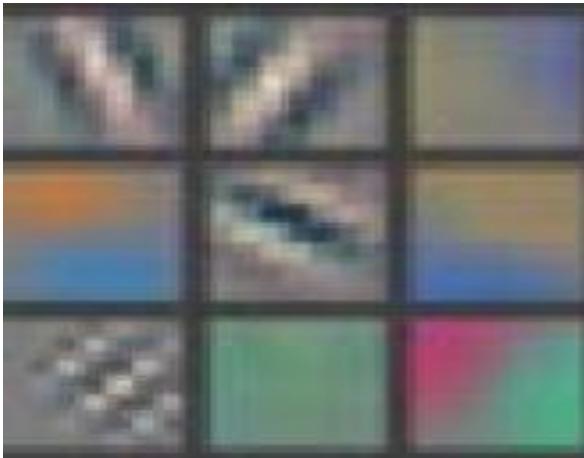
Lernen von Merkmalshierarchien

- Beispiele gelernter Objektteile für Kategorisierung



Gelernte Visuelle Merkmale

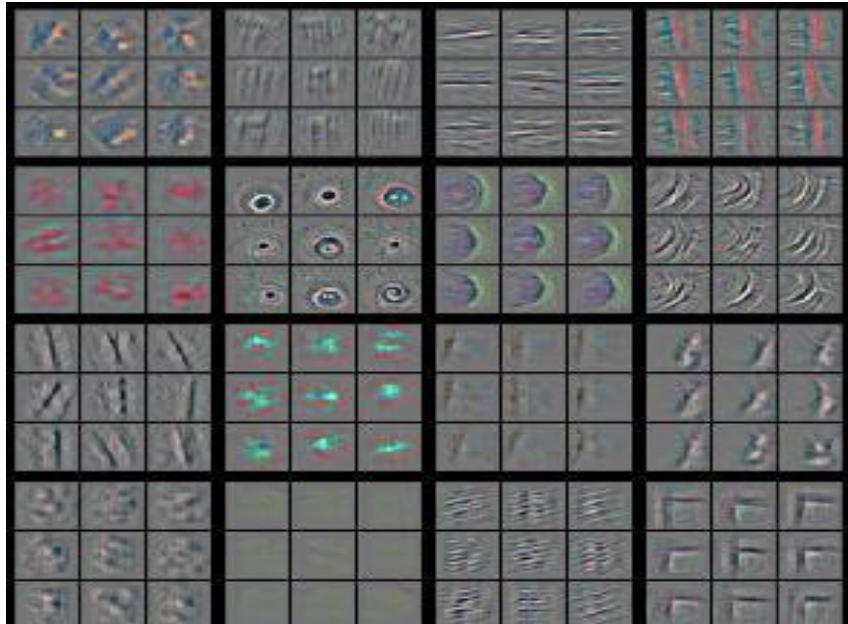
- Gewichte mit großem Beitrag zur Aktivierung
- Stark aktivierende Stimuli



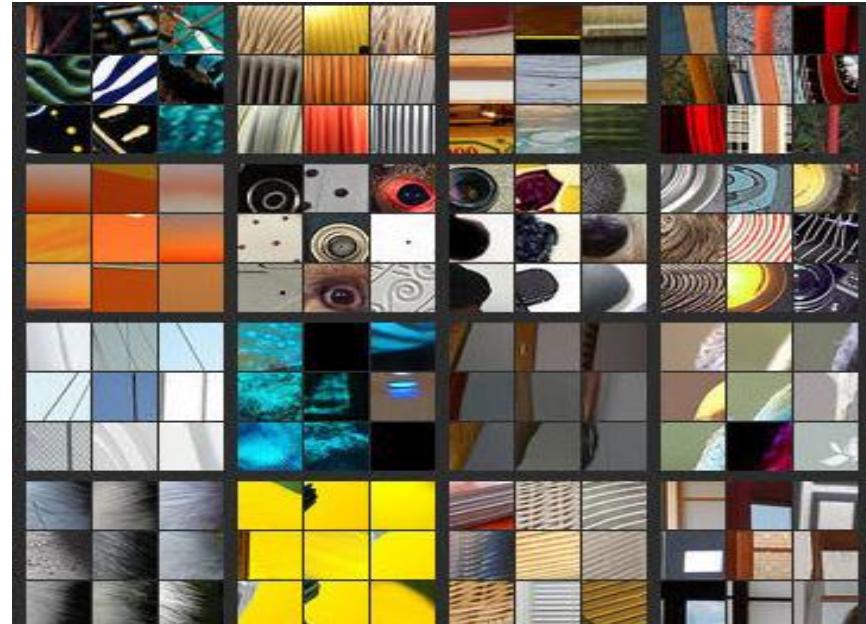
[Zeiler and Fergus 2014]

Gelernte Visuelle Merkmale

■ Eingabesensitivität



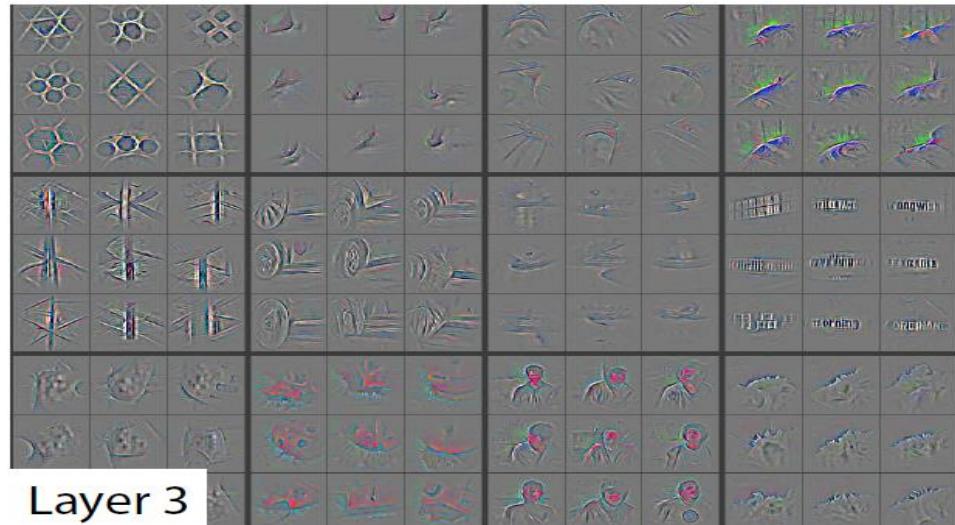
■ Stark aktivierende Stimuli



[Zeiler and Fergus 2014]

Gelernte Visuelle Merkmale

■ Eingabesensitivität



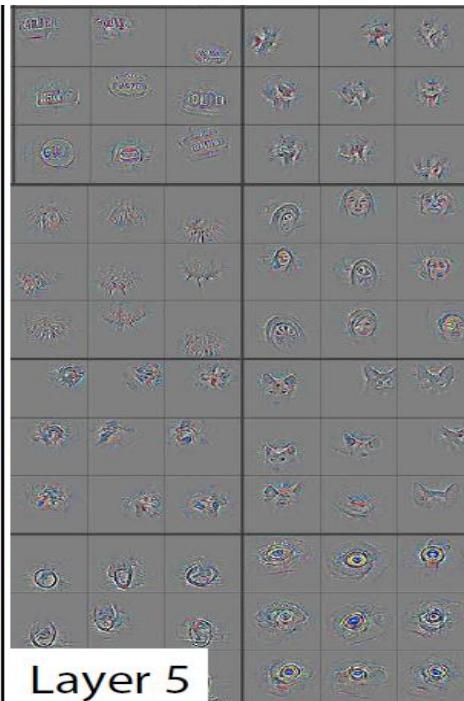
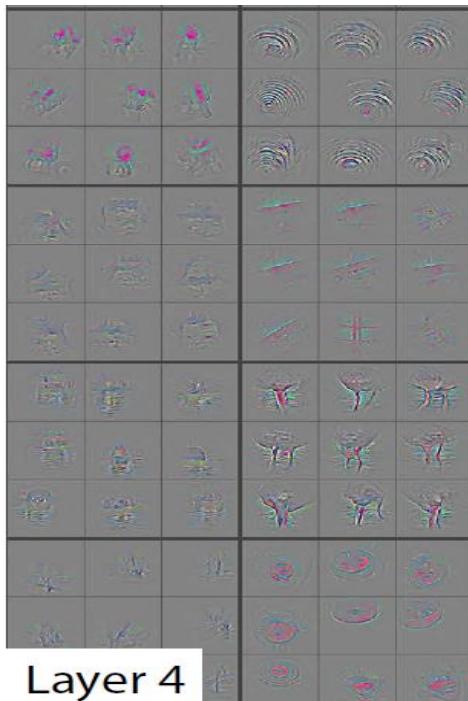
■ Stark aktivierende Stimuli



[Zeiler and Fergus 2014]

Gelernte Visuelle Merkmale

Eingabesensitivität und stark aktivierende Stimuli

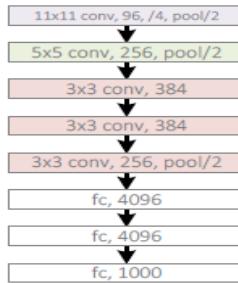


[Zeiler and Fergus 2014]

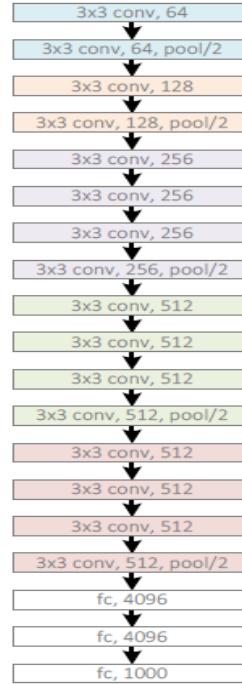
Beispiele von CNN-Strukturen der ILSVRC-Gewinner

Netzwerke immer tiefer

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)

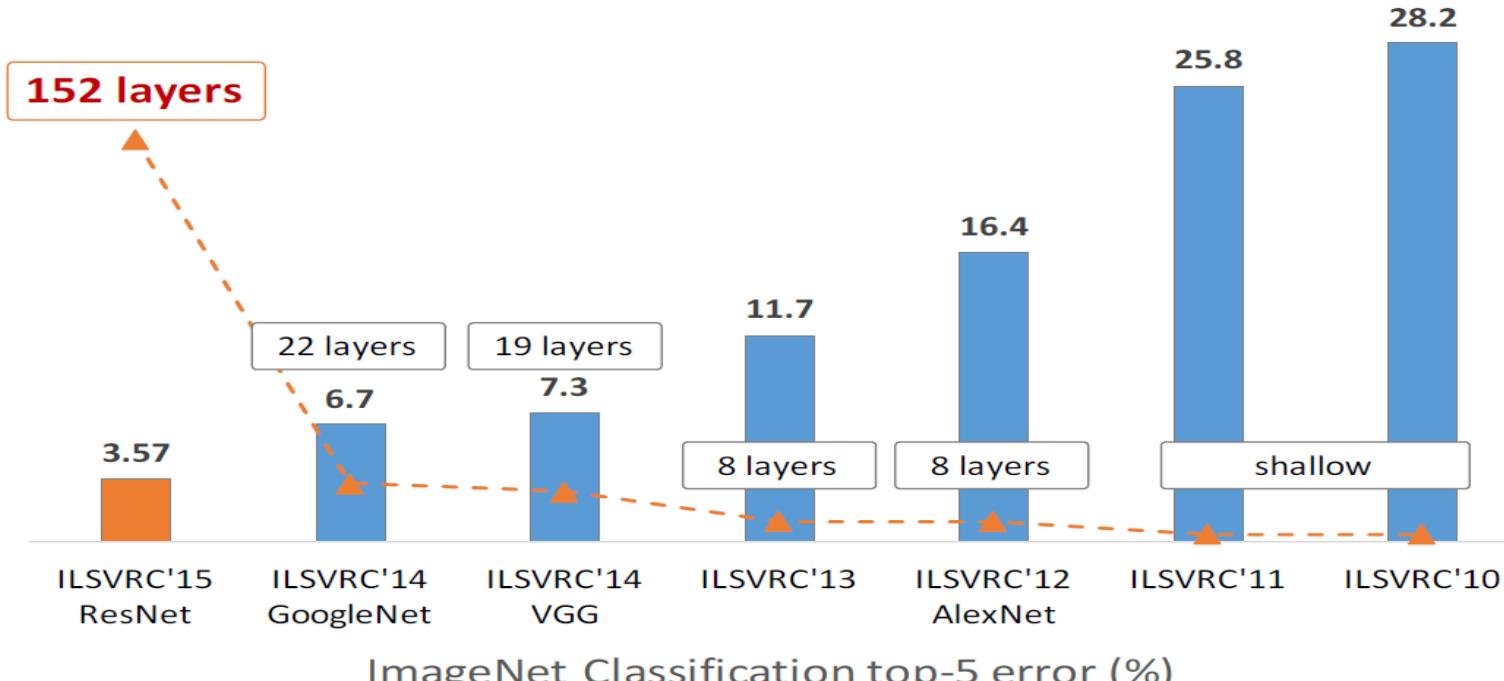


GoogleNet, 22 layers
(ILSVRC 2014)



[He CVPR 2016]

ImageNet-Objekterkennungsperformanz



[He et al. 2015]

Erkennung besser als Menschen



GT: horse cart

- 1: horse cart
- 2: minibus
- 3: oxcart
- 4: stretcher
- 5: half track



GT: birdhouse

- 1: birdhouse
- 2: sliding door
- 3: window screen
- 4: mailbox
- 5: pot



GT: forklift

- 1: forklift
- 2: garbage truck
- 3: tow truck
- 4: trailer truck
- 5: go-kart



GT: letter opener

- 1: drumstick
- 2: candle
- 3: wooden spoon
- 4: spatula
- 5: ladle

Top-5 Classification



GT: coucal

- 1: coucal
- 2: indigo bunting
- 3: lorikeet
- 4: walking stick
- 5: custard apple



GT: komondor

- 1: komondor
- 2: patio
- 3: llama
- 4: mobile home
- 5: Old English sheepdog



GT: yellow lady's slipper

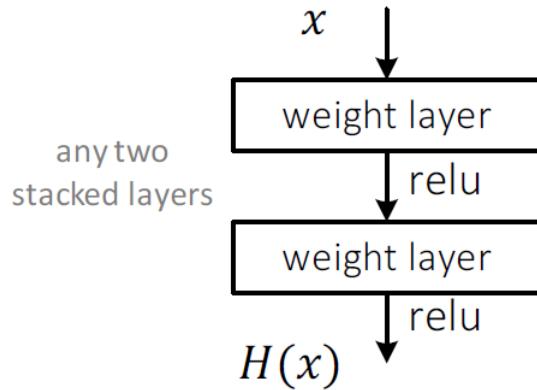
- 1: yellow lady's slipper
- 2: slug
- 3: hen-of-the-woods
- 4: stinkhorn
- 5: coral fungus



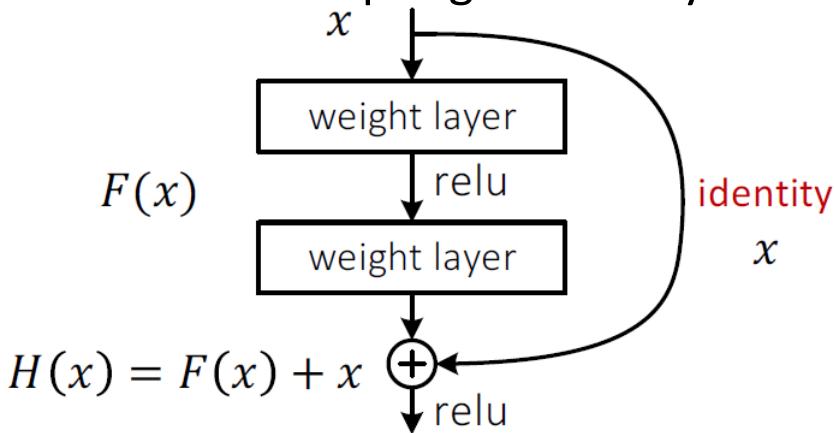
[He et al. 2015]

Deep Residual Learning

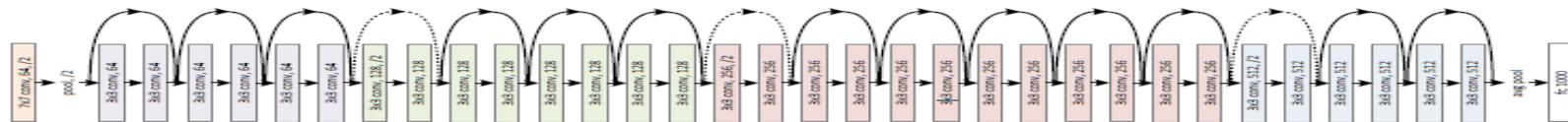
Geschichtetes Netz



Netz mit Überspringen von Layern



ResNet (Residual network): Sehr viele Schichten



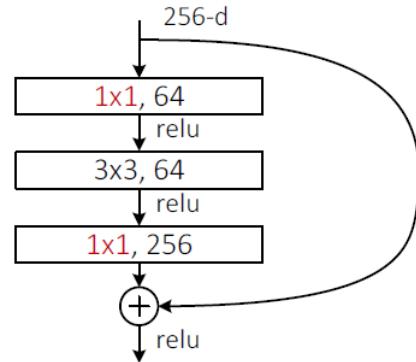
[He et al. CVPR 2016]

Iterative Verbesserung der Interpretation

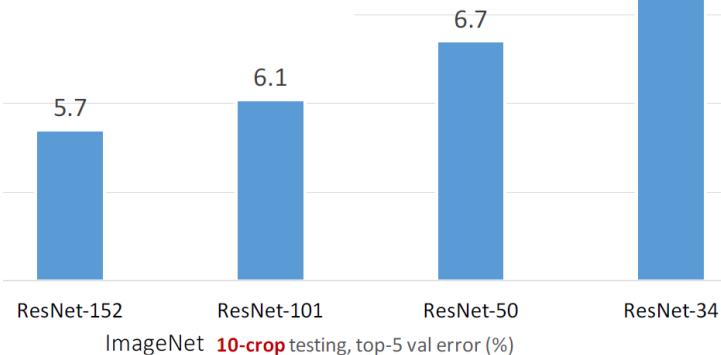
[Greff et al. ICLR 2017]

Lokale Engpässe

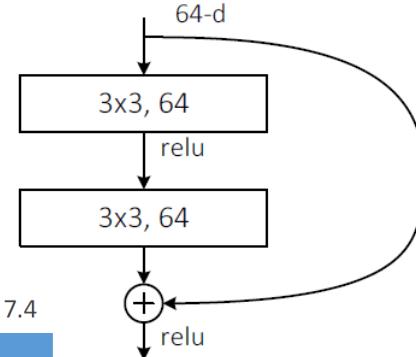
Engpass



Ähnliche
Komplexität



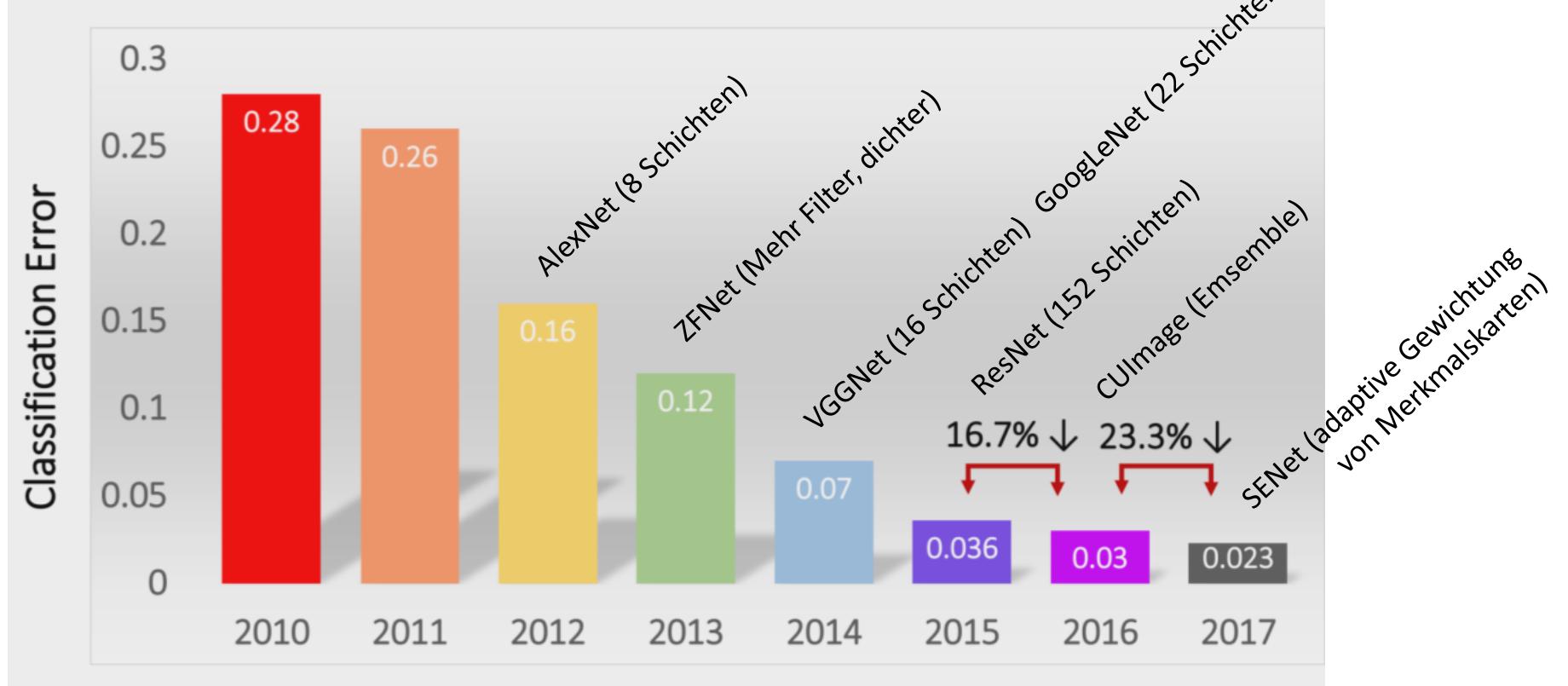
Zwei 3×3 Konvolutionen



[He et al. CVPR 2016]

Imagenet-Challenge (ILSVRC)

■ Jährliche Verbesserung der Bildkategorisierung



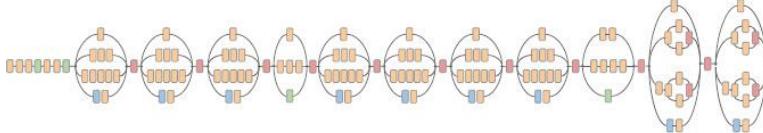
Erkennung von Hautkrebs

CNN-Training auf 129.450 Bildern

Skin lesion image



Deep convolutional neural network (Inception v3)



Training classes (757)

- Acral-lentiginous melanoma
- Amelanotic melanoma
- Lentigo melanoma
- ...
- Blue nevus
- Halo nevus
- Mongolian spot
- ...

Inference classes (varies by task)

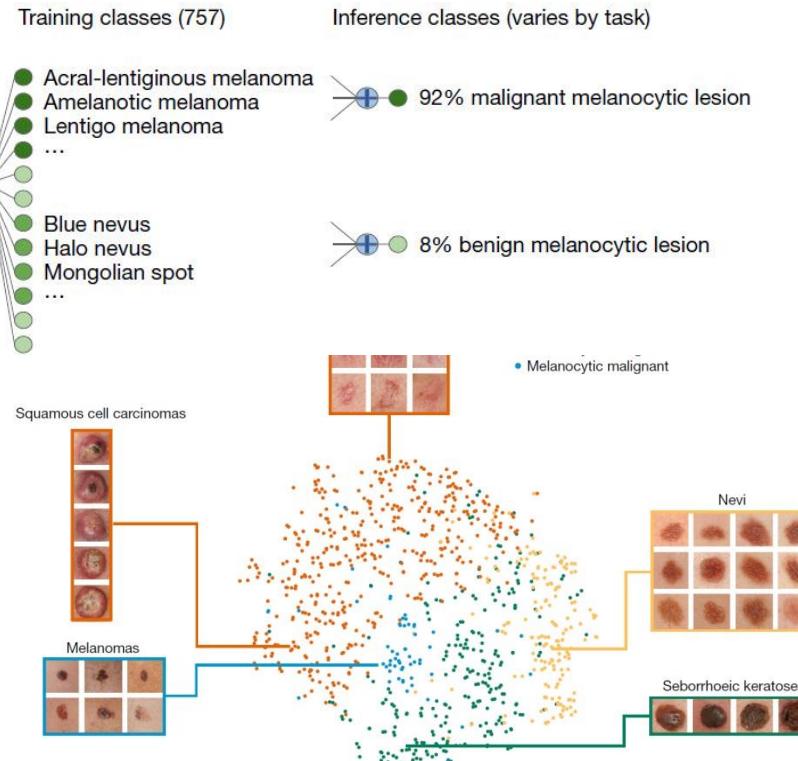
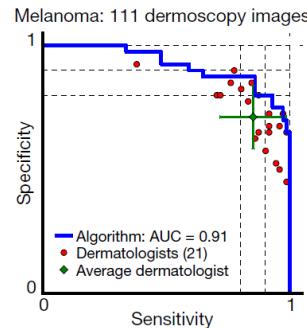
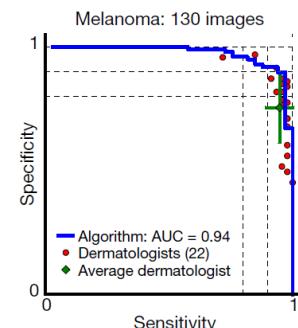
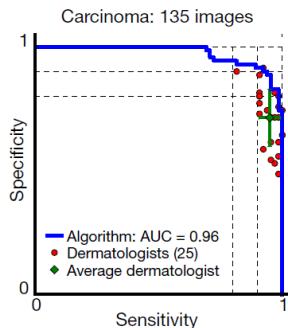
92% malignant melanocytic lesion

8% benign melanocytic lesion

Melanocytic malignant

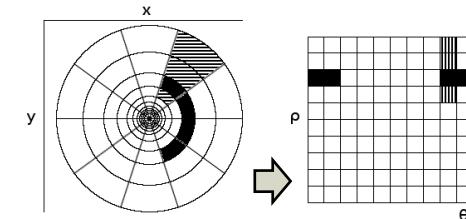
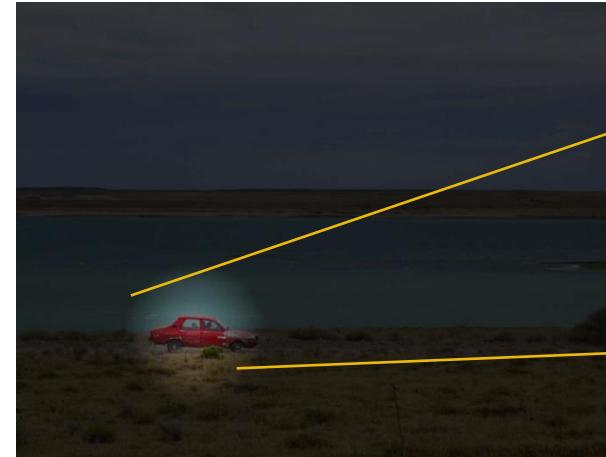
Gelernte Merkmale

Vergleich zu Dermatologen



Beschränkungen Konvolutionaler Verarbeitung

- Alle Bildpositionen werden auf gleiche Weise verarbeitet
=> mache dem Netzwerk Ortsinformation zugänglich
- Keine Invarianz gegenüber Skalierung, Drehung
=> Log-polare Vorverarbeitung
- Kein Fokus der Aufmerksamkeit
=> Hinschauen / Ausschneiden von Regionen (ROI)

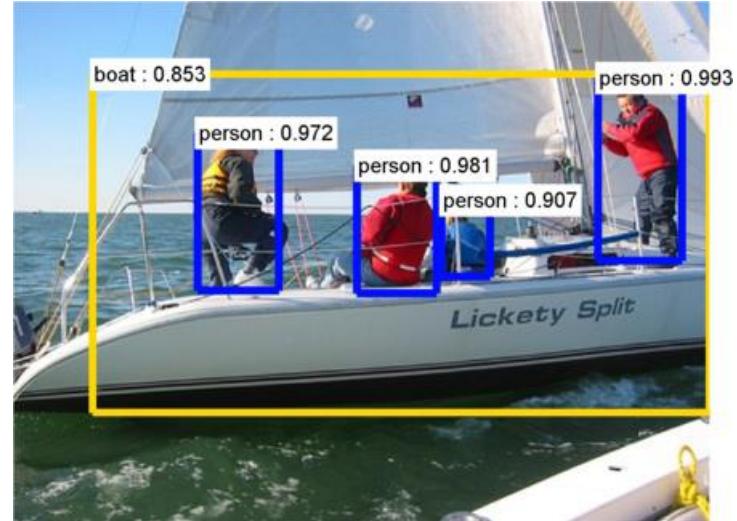


Objektdetektion

- Kategorisierung von Bildern
Was?



- Objektdetektion
Was + wo?



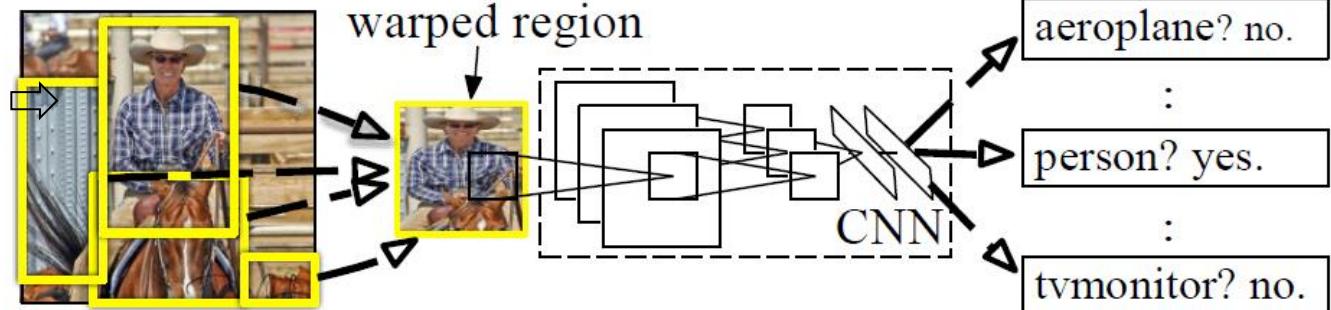
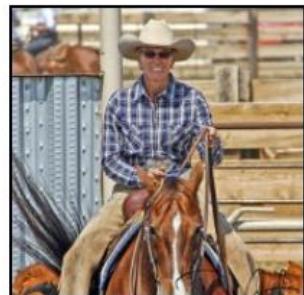
Beispiel Objektdetektion

- Annotation mit achsenparallelen Rechtecken (Bounding Boxes)
- Ortsabhängige Klassenwahrscheinlichkeiten auf mehreren Skalen
- Vorhersage der Bounding Boxes mit größter Klassenwahrscheinlichkeit
- Maximiere Überlappung vorhergesagter Bounding Boxes mit der Ground Truth
- Evaluation mit zwei Klassen der Pascal VOC 2007 -Datenmenge: Kühe und Pferde



Regionen-basierte CNN-Pipeline (R-CNN)

- Generiere Regionen für Objekthypothesen, z.B. anhand Farbe, Textur, ...
- Ausschneiden der Rols und Normalisierung der Größe
- Klassifikation der normalisierten Rols durch Konvolutionales Netz (CNN)



input image

region proposals
~2,000

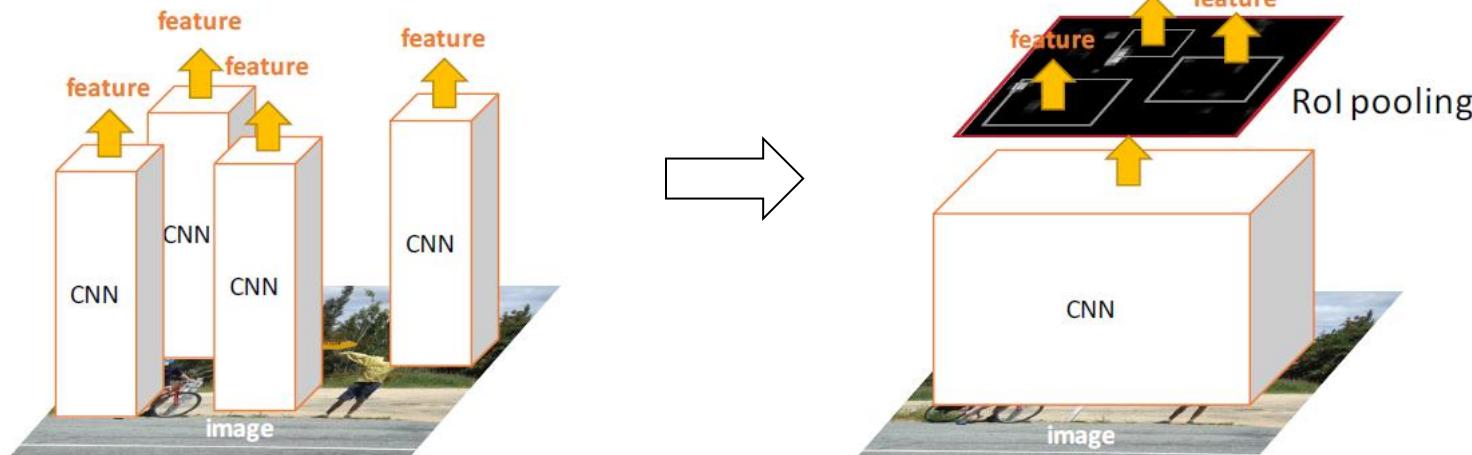
1 CNN for each region

classify regions

[Girshick et al. CVPR 2014]

Fast R-CNN

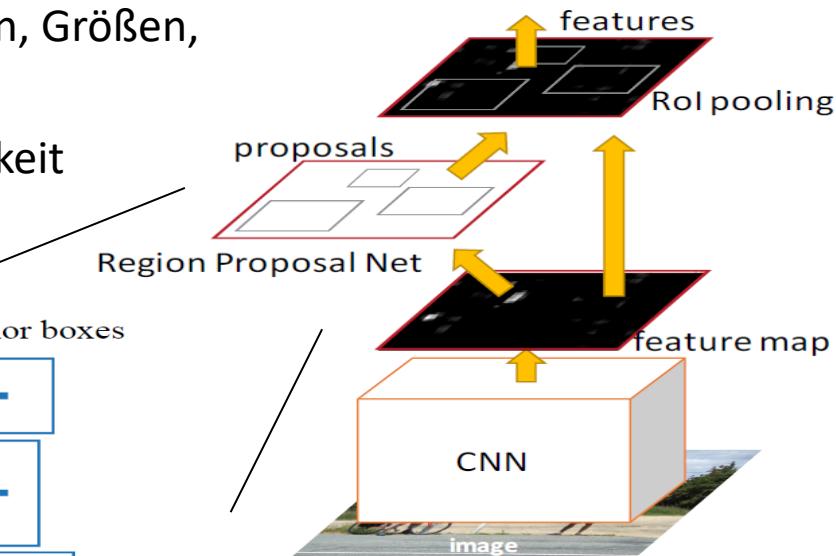
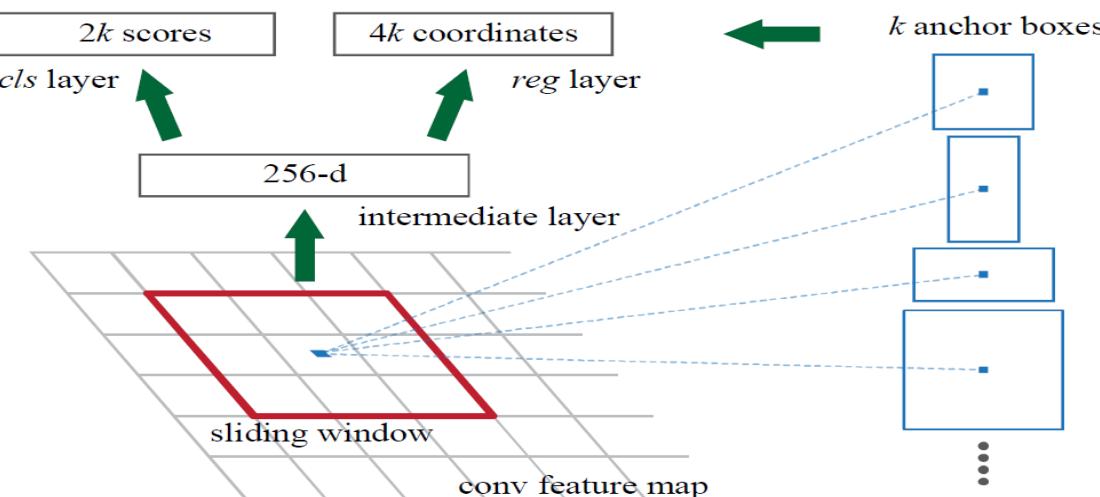
- Konvolutionale Berechnung für zahlreiche überlappende Regionen ineffizient
- Gemeinsame Berechnung der konvolutionalen Schichten und Ausschneiden der Merkmale (Region of interest pooling)



[Girschik ICCV 2015]

Faster R-CNN

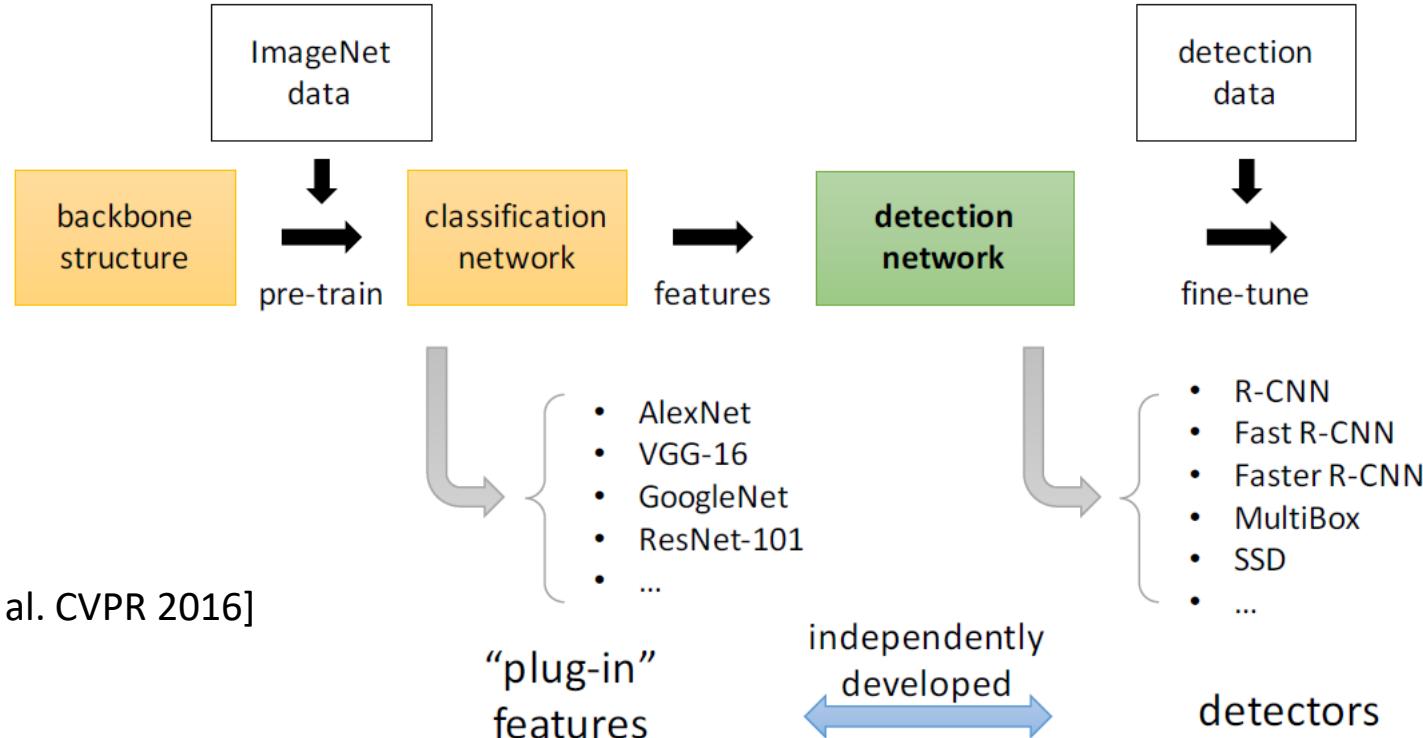
- Berechne Regionenvorschläge durch CNN
- Anker-Regionen in regelmäßigen Gitter von Orten, Größen, Seitenverhältnissen
- Pro Region Vorhersage von Objektwahrscheinlichkeit und relativen Koordinaten der Bounding Box



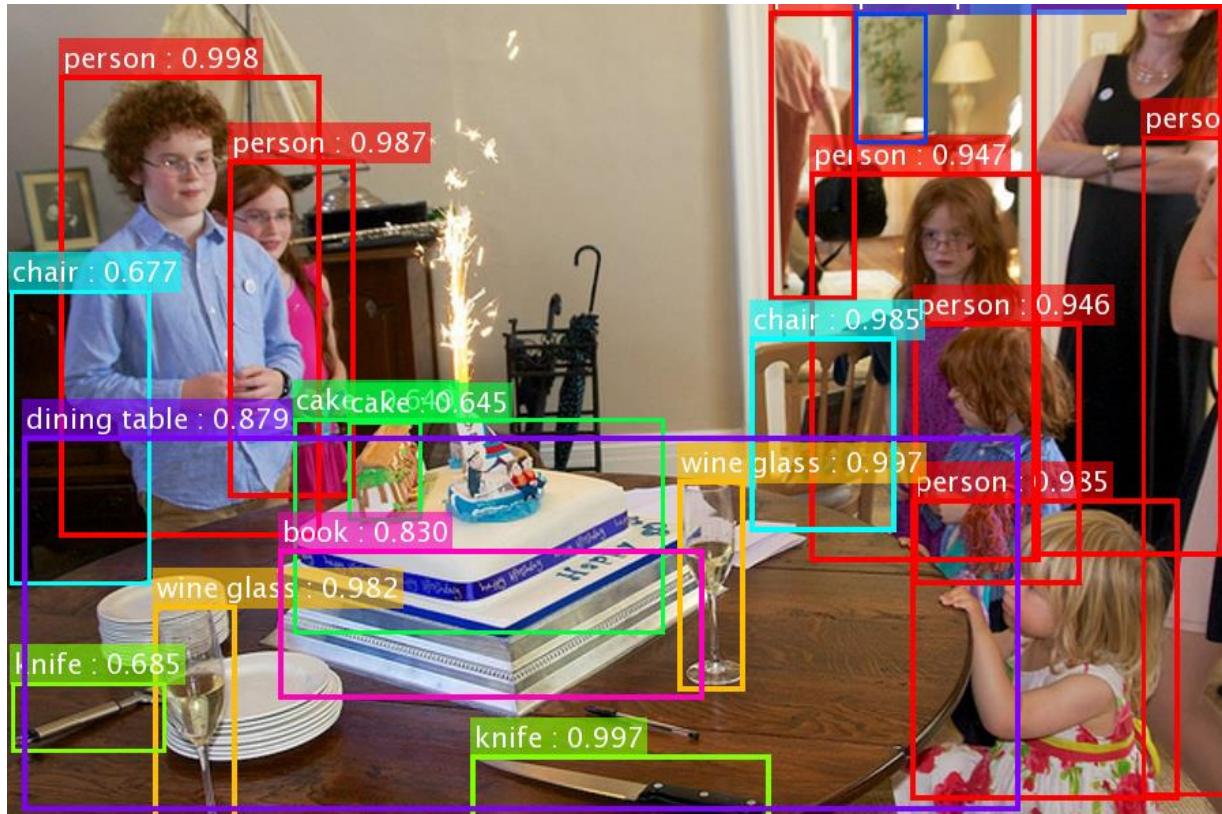
[Ren et al. NIPS 2015]

Objektdetections-Pipeline

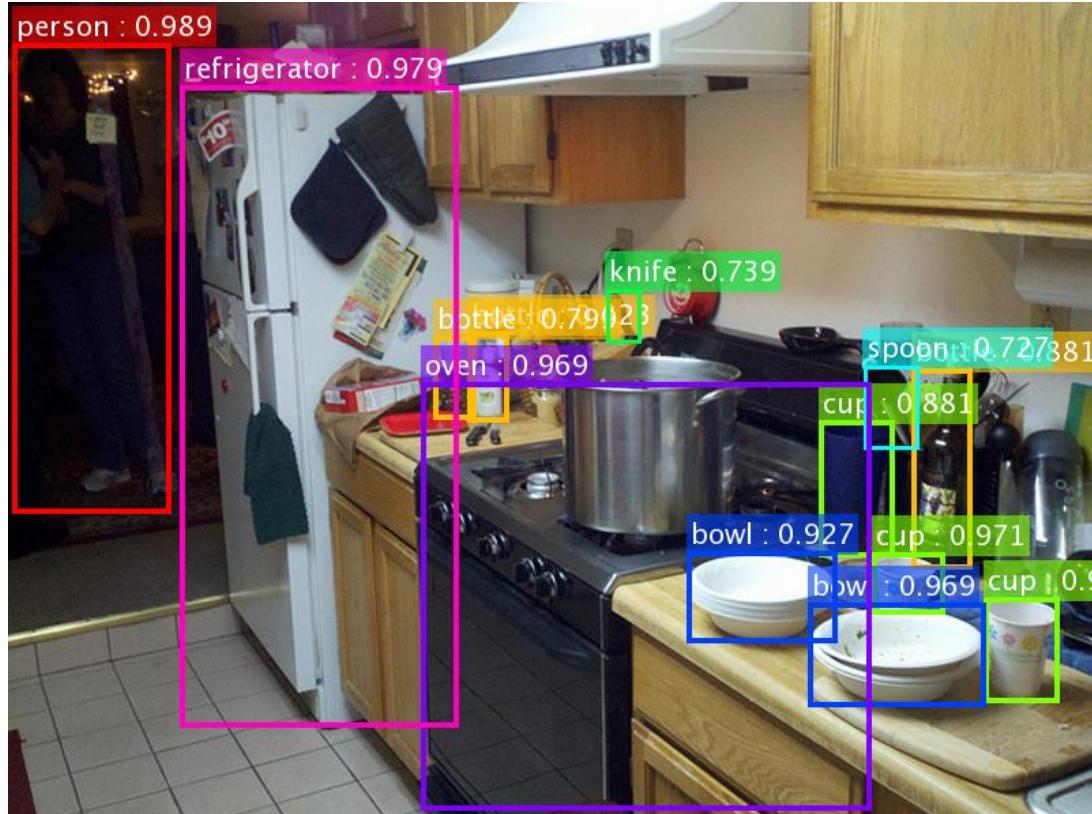
- Kombination von Kategorisierungs- und Detektionsmodellen
- Nutzung vortrainierter Merkmale (Transfer-Lernen)



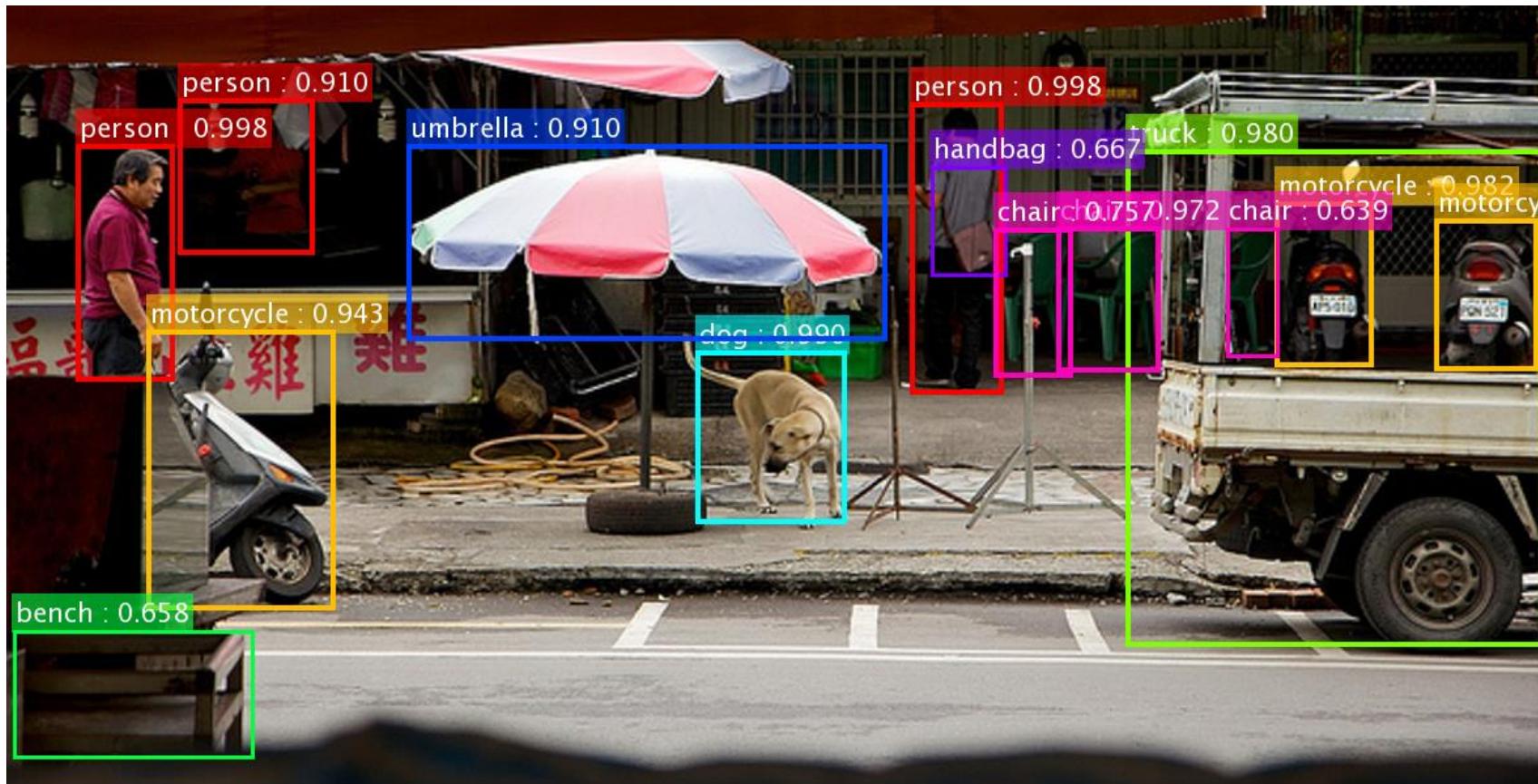
Faster R-CNN + ResNet Objektdetektionsergebnis



Faster R-CNN + ResNet Objektdetektionsergebnis

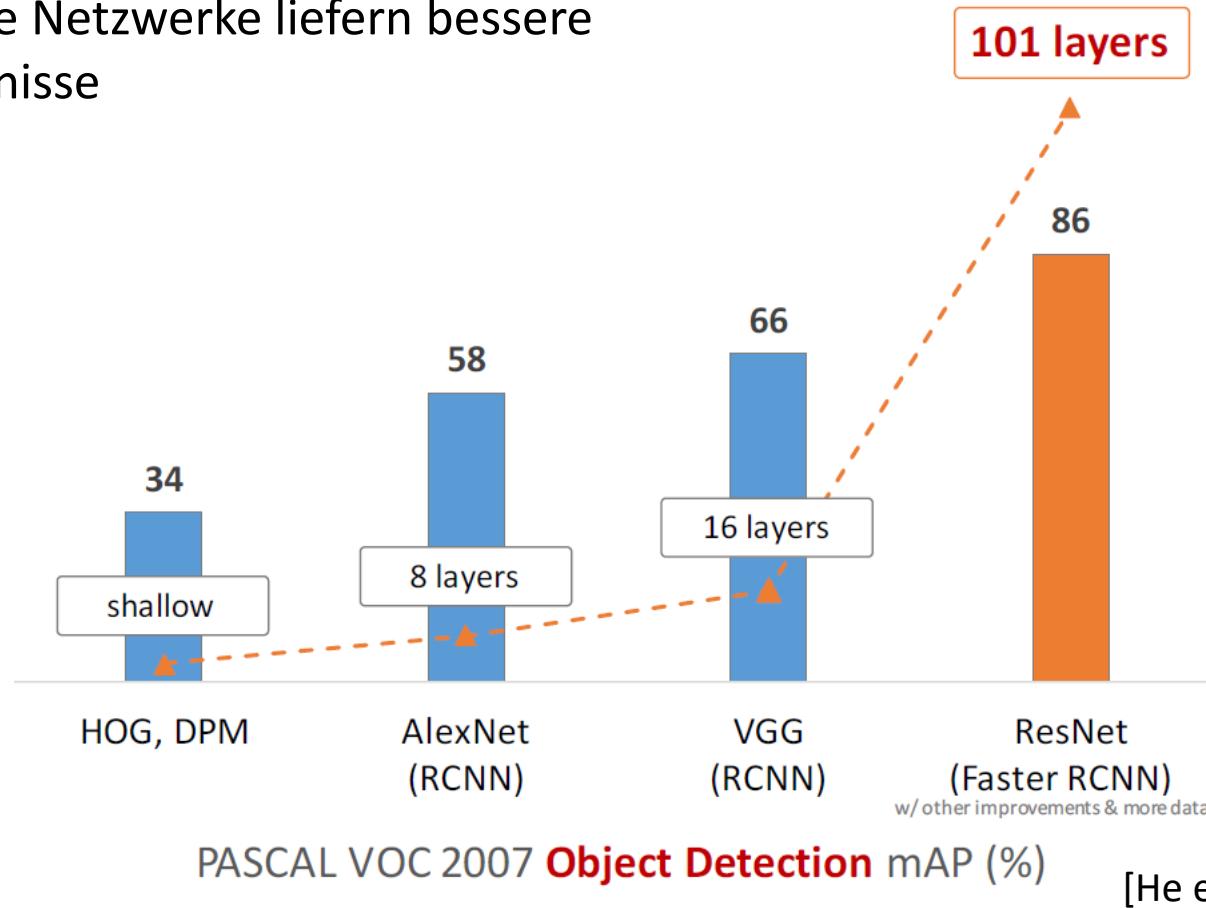


Faster R-CNN + ResNet Objektdetektionsergebnis

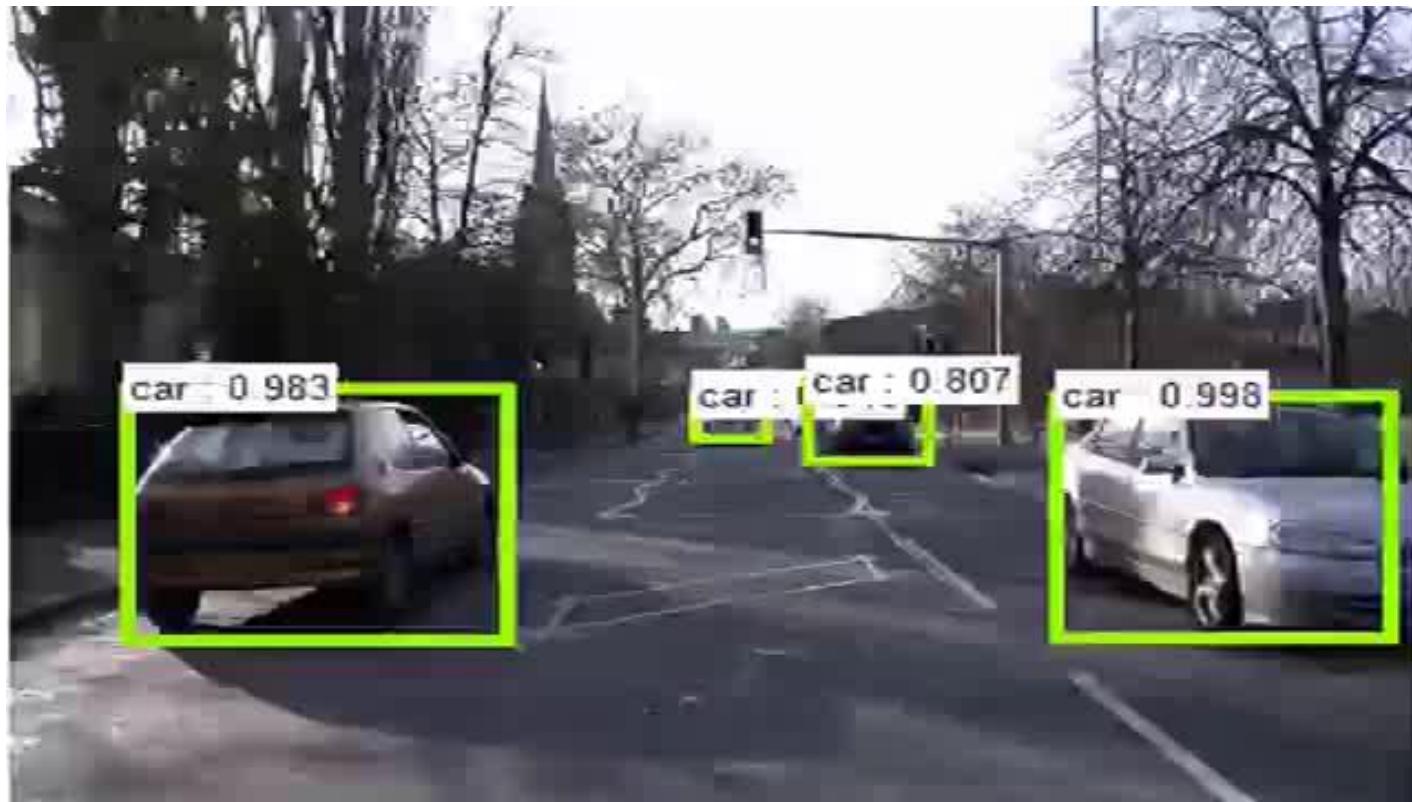


Objektdetections-Performanz

- Tiefere Netzwerke liefern bessere Ergebnisse

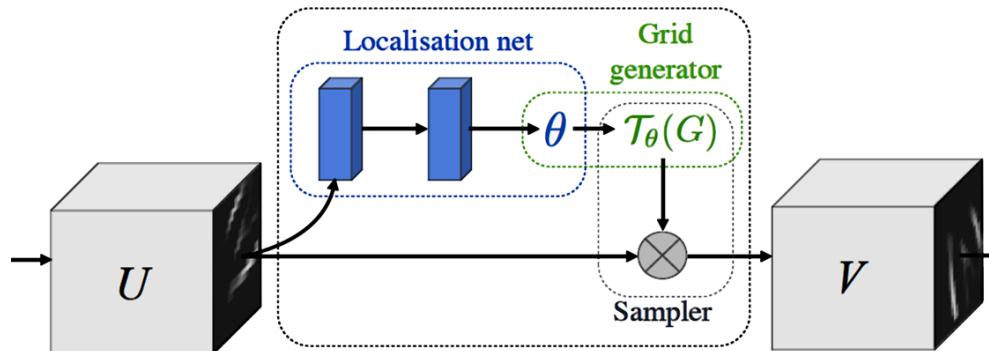
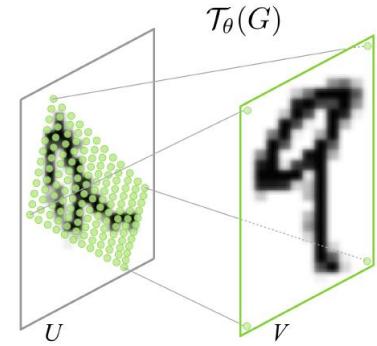


Faster R-CNN + ResNet Objektdetektion in Video

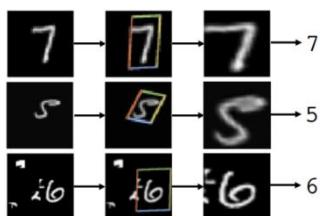


Spatial Transformer Networks

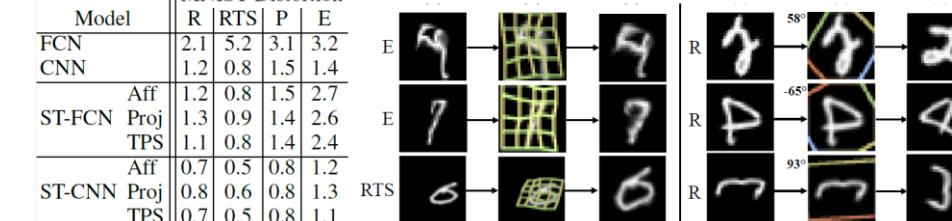
- Definiere parametrisierte Bildtransformation, z.B. affin
- Lokalisierungsnetzwerk schätzt Transformationsparameter θ
- Grid-Generator berechnet Ursprungskoordinaten im Bild
- Sampler führt die Transformation aus => Normalisierte Rol



[Jaderberg et al. NIPS 2015]



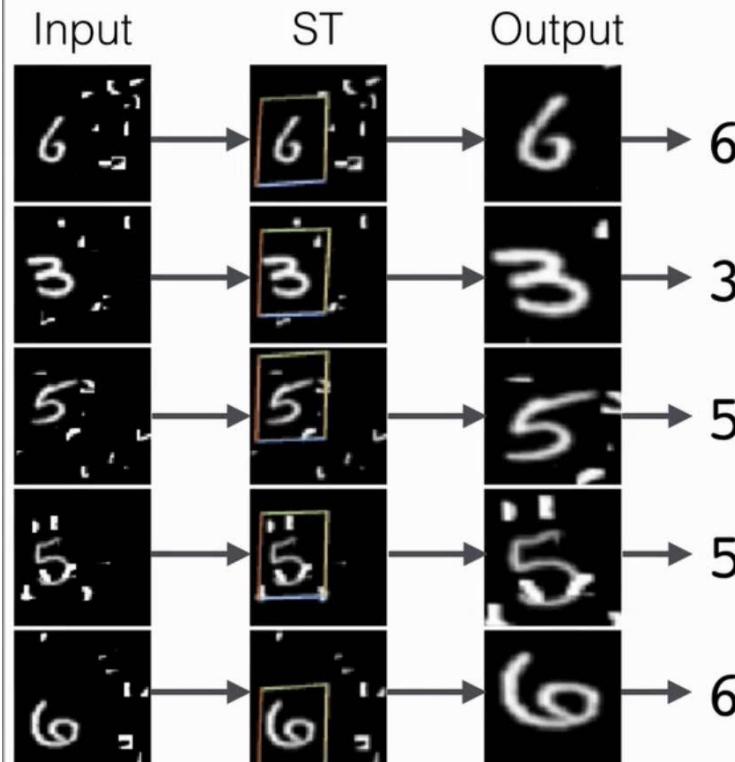
Model	MNIST Distortion				
	R	RTS	P	E	
FCN	2.1 1.2	5.2 0.8	3.1 1.5	3.2 1.4	
CNN					
ST-FCN	Aff Proj TPS	1.2 1.3 1.1	0.8 0.9 0.8	1.5 1.4 1.4	2.7 2.6 2.4
ST-CNN	Aff Proj TPS	0.7 0.8 0.7	0.5 0.6 0.5	0.8 0.8 0.8	1.2 1.3 1.1



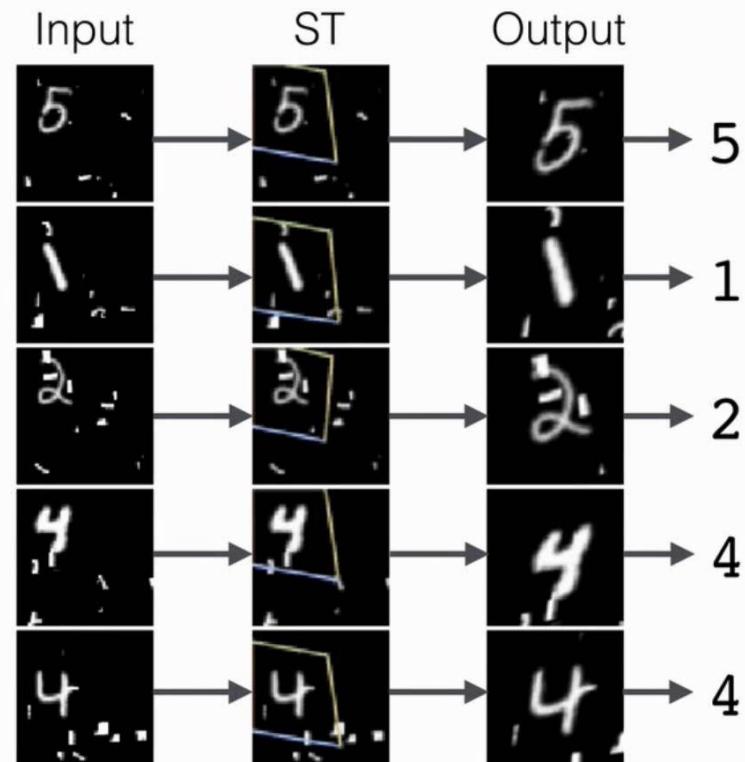
Spatial Transformer Networks

Translated Cluttered MNIST

ST-FCN Affine



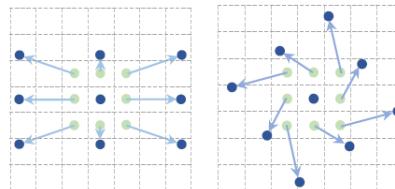
ST-CNN Affine



[Jaderberg et al. NIPS 2015]

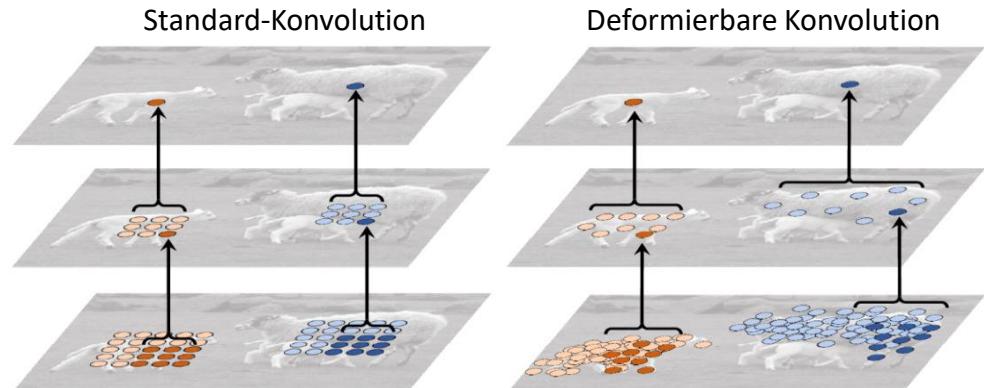
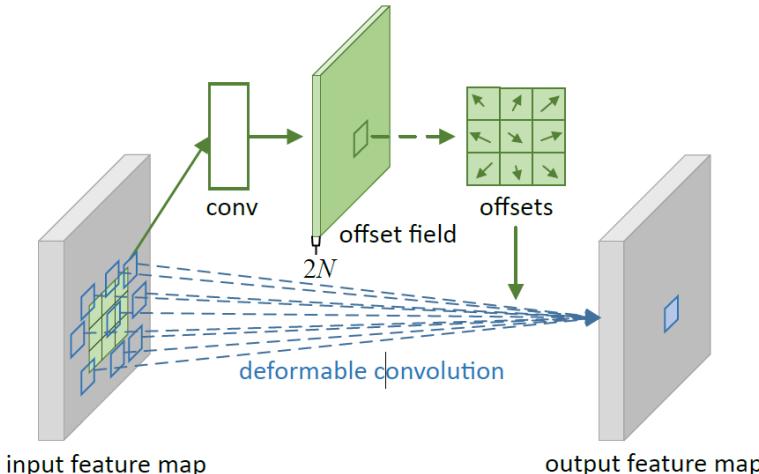
Deformierbare Konvolutionale Netzwerke

- Ähnliche Idee wie bei Spatial Transformer Networks, aber lokal in einer CNN-Schicht
- Lokale Deformation auf mehreren Ebenen



=> Rezeptive Felder passen sich flexibel an die Eingabe an

[Dai et al. 2017]



Deformierbare Konvolutionale Netzwerke

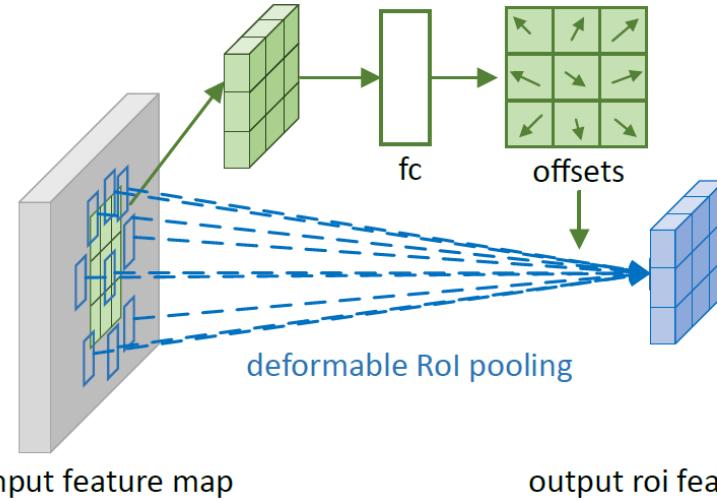
- Rezeptives Feld (rote Punkte) für Objekt an grünem Punkt



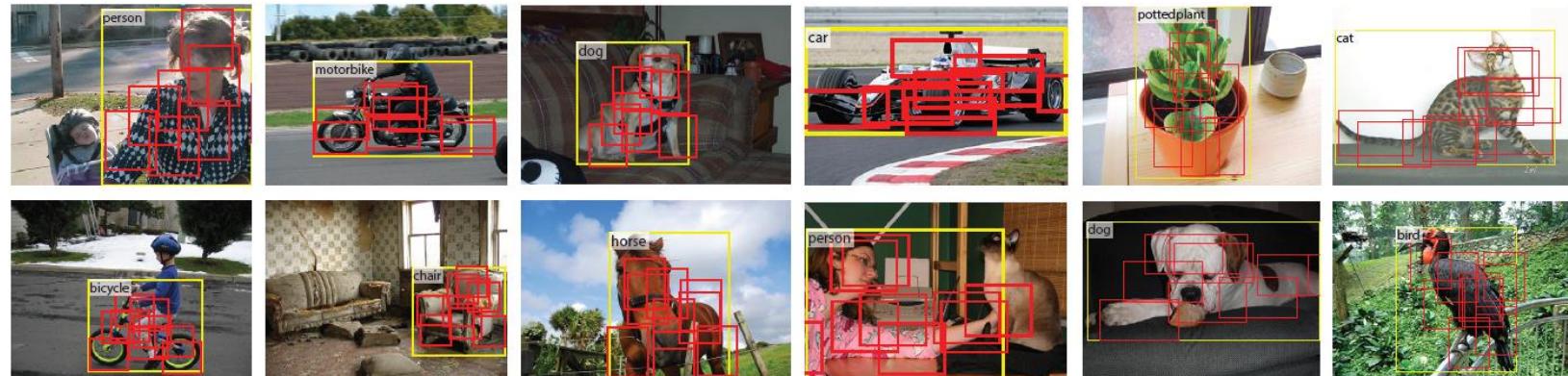
[Dai et al. 2017]

Deformierbare Konvolutionale Netzwerke: Teile

- Deformierbares teilebasiertes
RoI-Pooling nach
konvolutionalen Schichten
um Objekt auszuschneiden
- Platzierung der Objektteile
wird Eingabe angepasst



[Dai et al. 2017]



Semantische Segmentierung

- Ordne jeden Bildpunkt einer Klasse zu

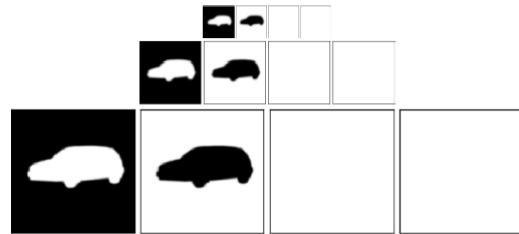


[Neuhold, et al. 2017]

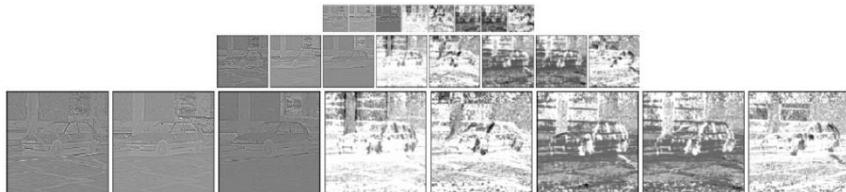
Semantische Segmentierung

[Schulz, Behnke 2012]

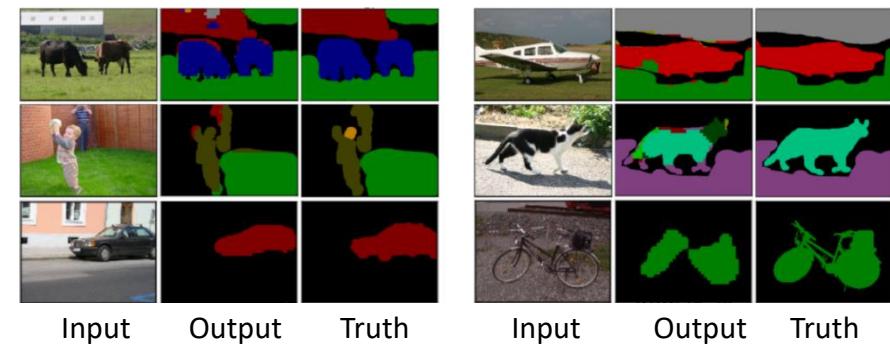
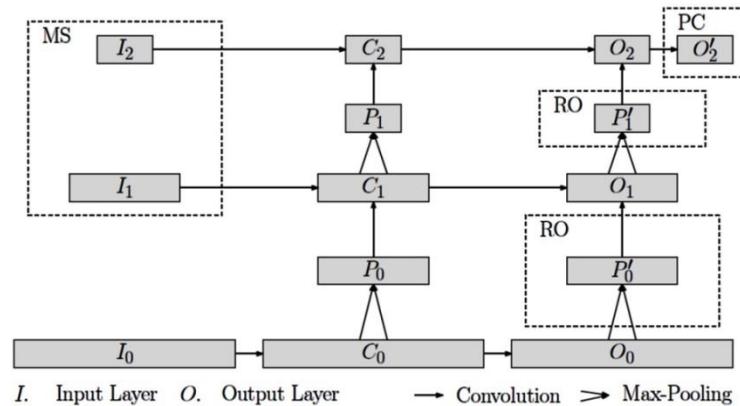
Multiresolutions-Targets



Multiresolutions-Eingaben

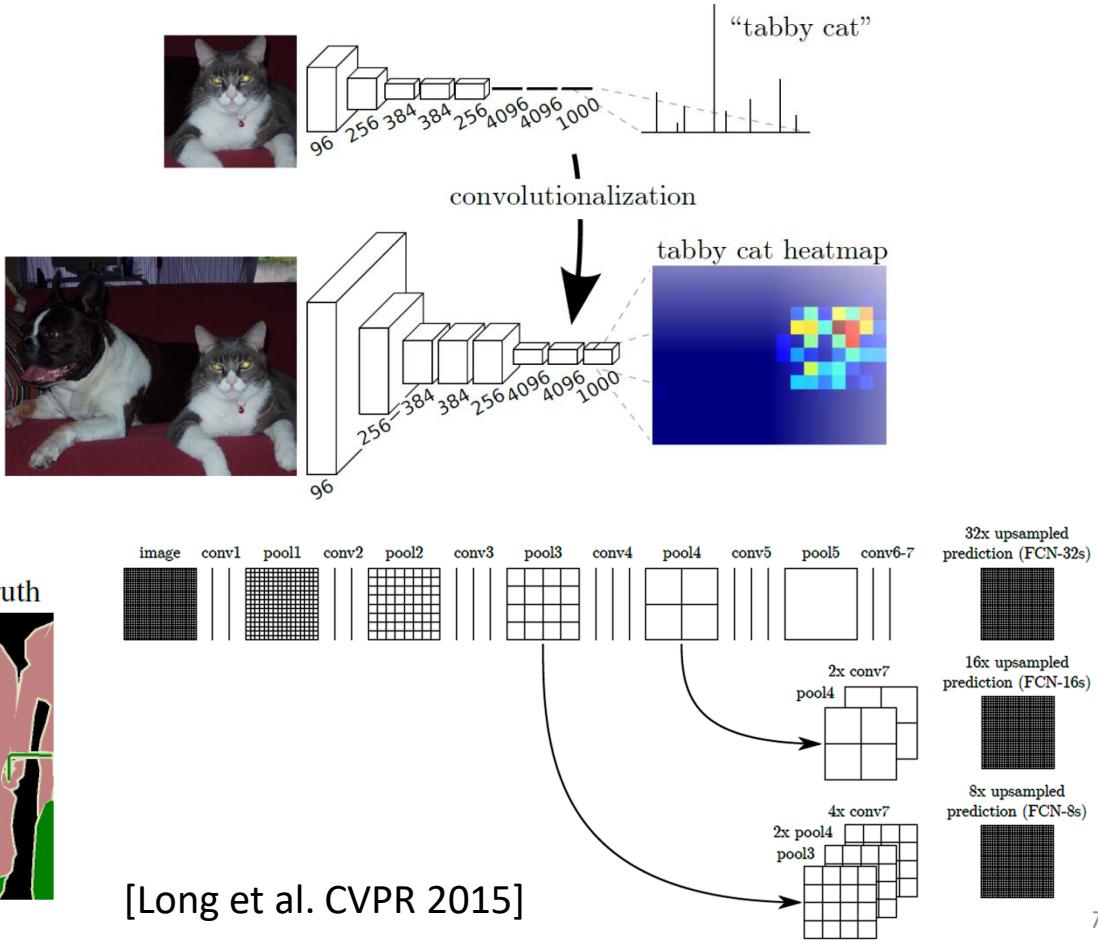
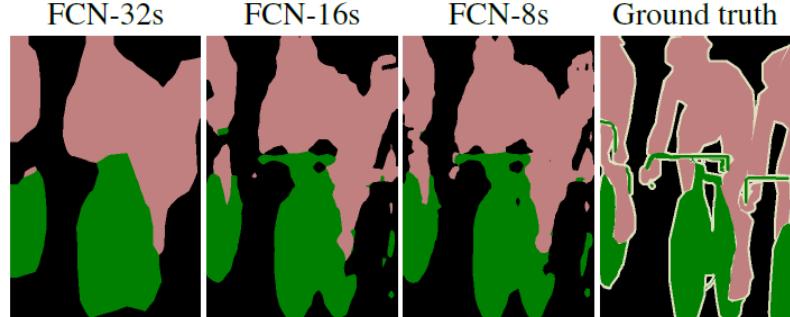


Evaluiert auf MSRC-9/21 und INRIA Graz-02 Daten



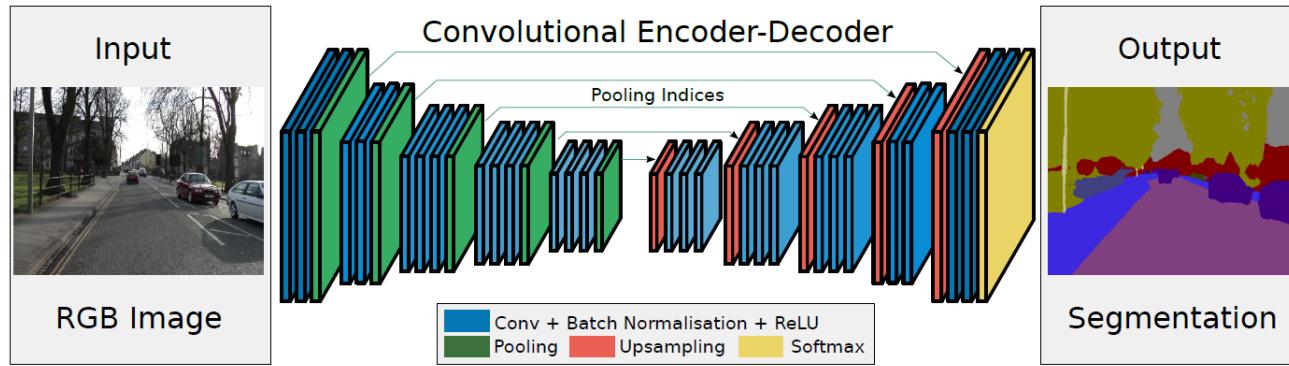
Fully Convolutional Networks

- Man könnte Klassifikationsnetzwerk an allen Pixeln auswerten
- Problem: Grobe Ausgabeauflösung
- Idee: Mache Pooling durch Upsampling rückgängig



SegNet: Encoder-Decoder

- Nutze Pooling-Indizes für Upsampling

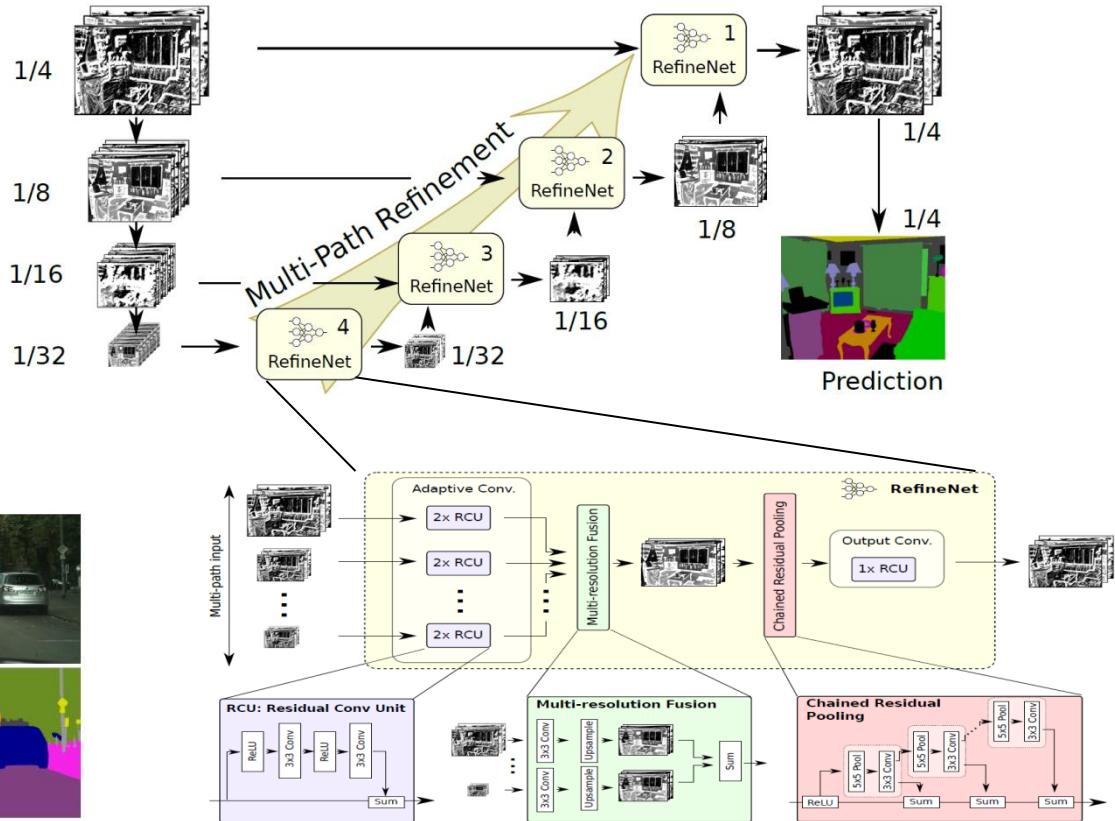


[Badrinarayanan et al. PAMI 2017]

RefineNet

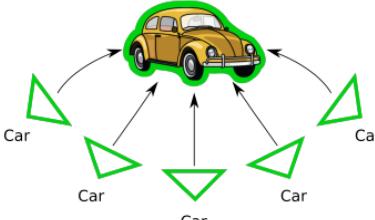
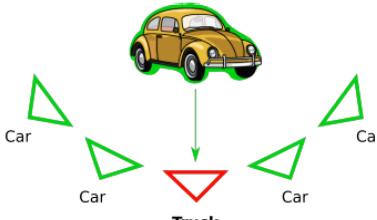
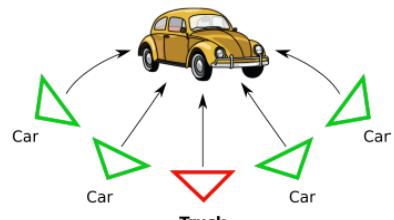
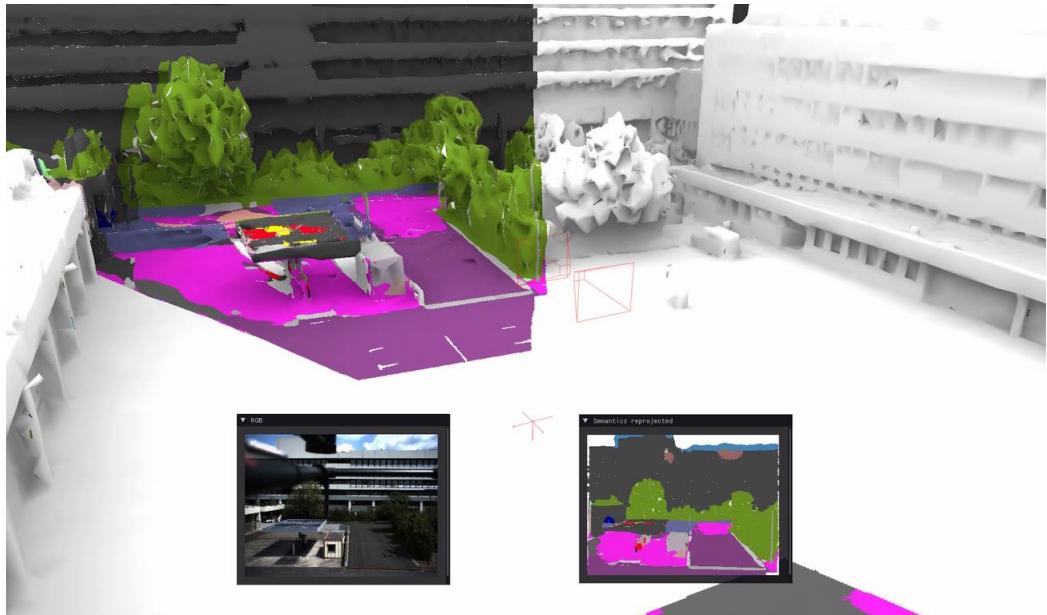
[Lin et al. CVPR 2017]

- Erhöhung der Auflösung durch Nutzung von Merkmalen in der höheren Auflösung
- Grob-zu-Fein-Strategie
- Objekt-Parsen und semantische Segmentierung



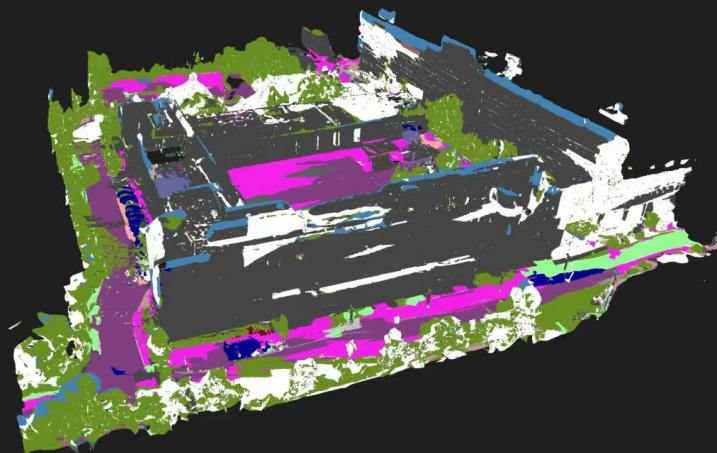
Semantische Kartierung

- Semantische Segmentierung von Bildern
- 3D-Kartierung durch Registrierung von Messungen eines 3D-Laserscanners
- Segmentierung wird als Textur für Mesh verwendet
- Fusion verschiedener Ansichten
- Label-Propagation



[Rosu et al., IJCV 2020]

Semantische Karte



[Rosu et al., IJCV 2020]

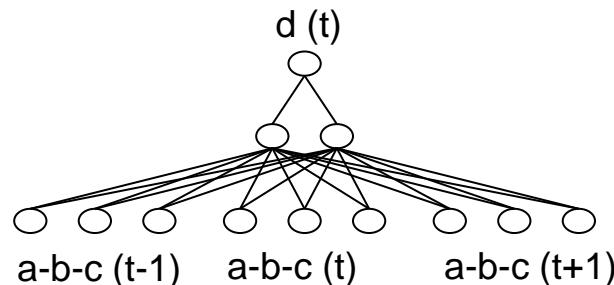
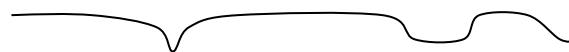
COMPUTATIONAL INTELLIGENCE **Rekurrente Netze**

Prof. Dr. Sven Behnke

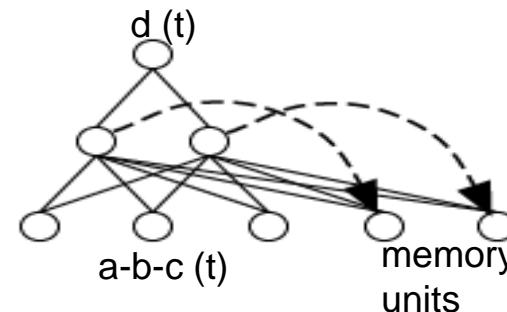
Verarbeitung von Zeitserien

■ Extraktion zeitabhängiger Merkmale

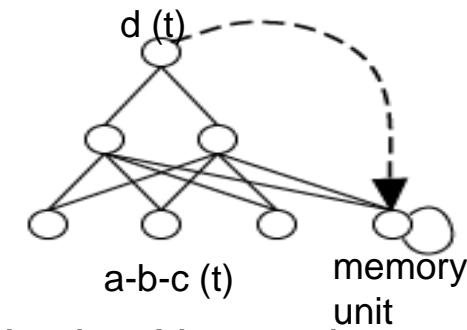
Ausgabe: d



Time Delay-Netzwerk

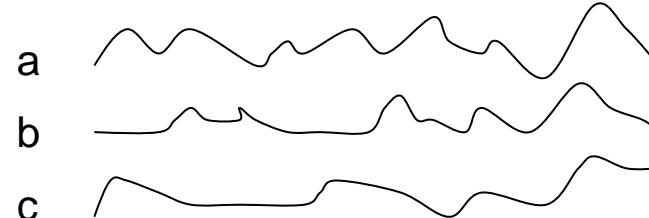


Elman-Netzwerk



Jordan-Netzwerk

Eingaben:

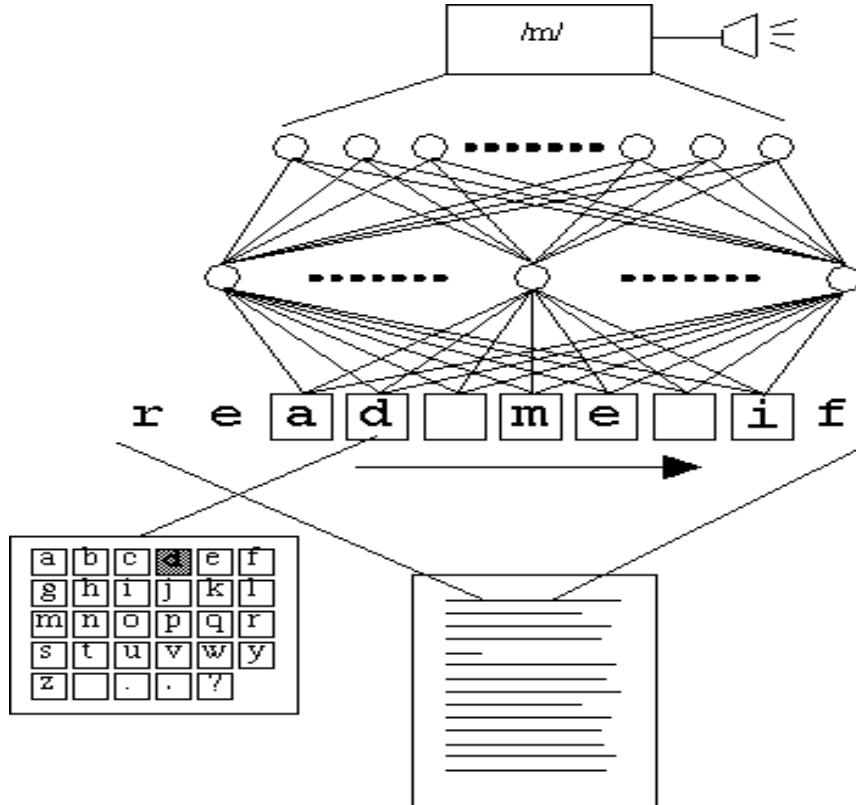


NETtalk

- Neuronales Netz lernt Aussprache von Text:

- 7x29 Eingabe-Units codieren Buchstaben in Eingabefenster (TDNN)
- 26 Ausgabe-Units codieren Phoneme
- Ca. 80 verdeckte Knoten

- Training auf Text mit 1000 Worten
- Test-Genauigkeit 95%



[Sejnowski & Rosenberg, 1987]

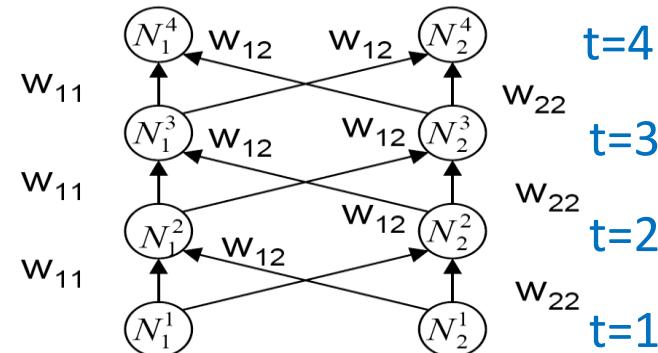
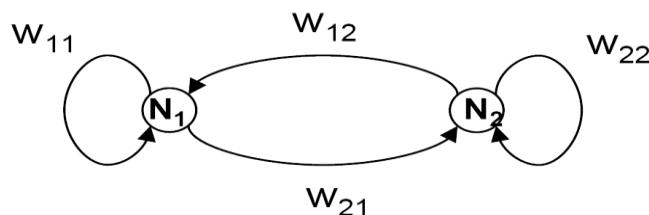


Rekurrente Netzwerke

- Wenn die Netzverbindungen gerichtete Zyklen enthalten, kann das Netz viel mehr berechnen, als nur kombinatorische Funktionen:
 - Es kann oszillieren
 - Gut für Bewegungssteuerung?
 - Es kann gegen Attraktoren konvergieren
 - Gut für Klassifikation
 - Es kann sich chaotisch verhalten
 - Normalerweise keine gute Idee für die Informationsverarbeitung
 - Es kann sich Dinge lange merken
 - Das Netzwerk hat internen Zustand. Es kann Informationen rekursiv filtern.
 - Es kann sequentielle Daten einfach modellieren
 - Zeitdimension muss nicht explizit modelliert werden

Backpropagation Through Time (BPTT)

- Methode zum Training rekurrenter Netze
- Entfalten entlang der Zeitachse
- Umwandlung in ein vorwärtsgerichtetes Netz

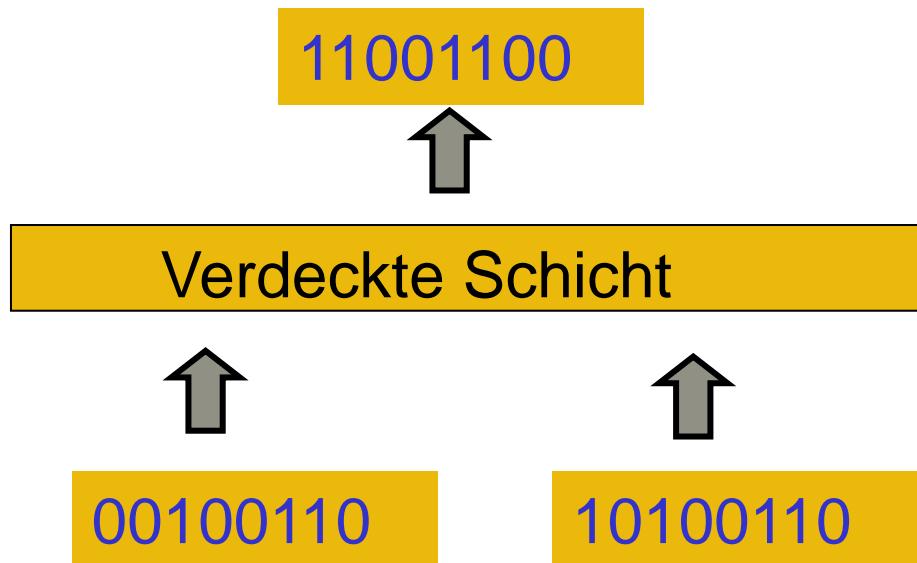


$$o_i(t+1) = f(\text{net}_i(t)) = f\left(\sum_j w_{ij} o_j(t) + x_i(t) \right)$$

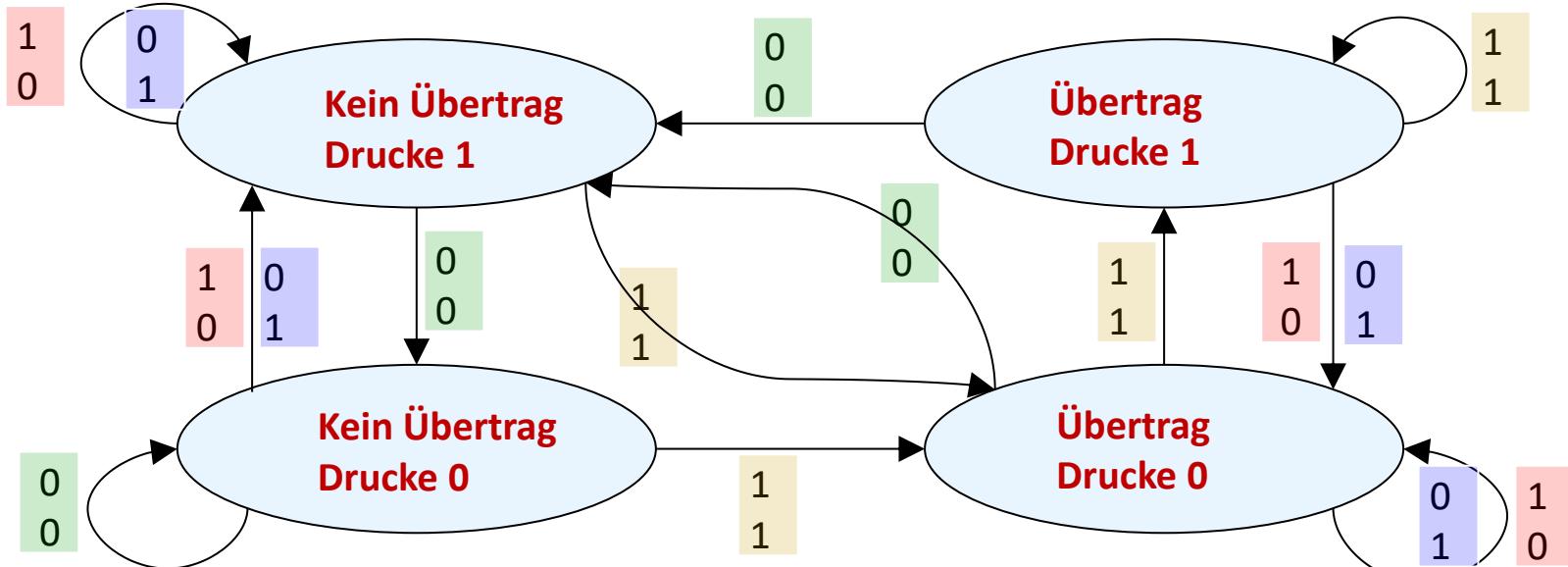
- Weight-sharing entlang der Zeitachse -> Updates mitteln
- Gewünschte Ausgabe konstant für Attraktoren
- Lerne auch Anfangsbelegungen der verdeckten Knoten

Beispiel-Problem mit Rekurrenz

- Man kann ein vorwärtsgerichtetes neuronales Netz trainieren, die binäre Addition zu lernen, aber es gibt Regularität, die das Netz nicht erfassen kann:
 - Die Anzahl der Bits der Zahlen muss vorher festgelegt werden
 - Die Verarbeitung am Anfang des Bitstrings generalisiert nicht auf die Verarbeitung am Ende, denn es werden verschiedene Gewichte verwendet
- Daher können vorwärts-gerichtete neuronale Netze die binäre Addition nicht gut lernen



Algorithmus für binäre Addition



- Dies ist ein endlicher Automat. Durch Betrachtung einer Spalte wird entschieden, welcher Zustandsübergang stattfindet. Dann wird ein Ausgabezeichen erzeugt. Die Verarbeitung erfolgt von niedrigwertig nach hochwertig.

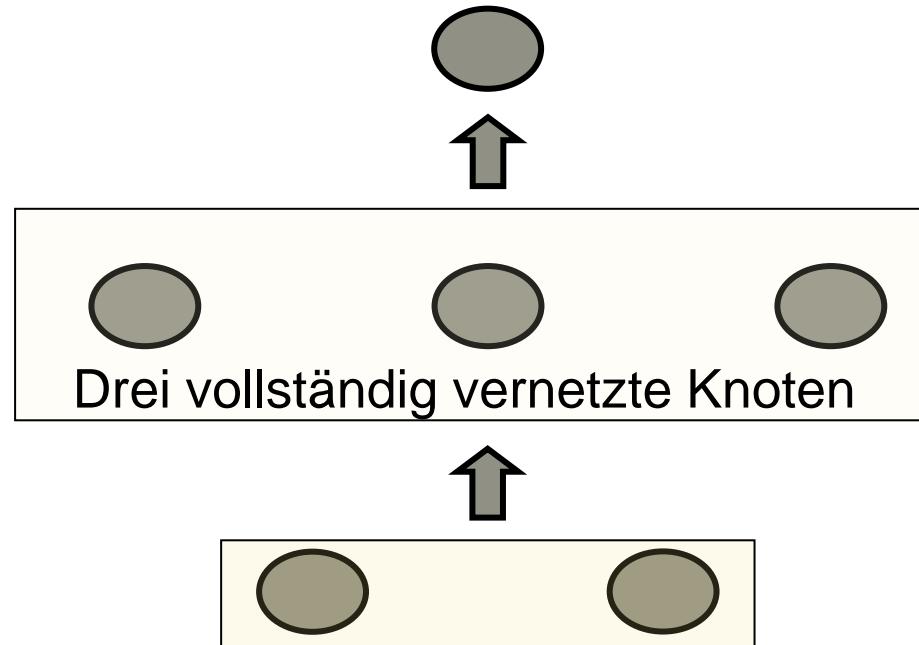
Rekurrentes Netz für binäre Addition

- Das Netzwerk hat zwei Eingabe-Units und eine Ausgabe-Unit
- Pro Zeitschritt sieht das Netz zwei Eingabe-Bits
- Die Zielausgabe ist jeweils die Ausgabe, die zur Eingabe vor zwei Zeitschritten gehört
 - Ein Zeitschritt wird für die Berechnung der Units in der verdeckten Schicht benötigt
 - Ein weiterer Schritt wird für die Berechnung der Ausgabe benötigt



Verbindungsstruktur des Netzes

- Die drei Knoten in der verdeckten Schicht sind vollständig vernetzt
 - Dies erlaubt die Berechnung eines Aktivitätsumusters in Abhängigkeit des vorigen Aktivitätsumusters
- Die Eingaben sind vollständig mit den Knoten der verdeckten Schicht verbunden
 - D.h. die Aktivität der verdeckten Knoten hängt auch von der Eingabe ab



Was das Netz lernt

- Es werden verschiedene Aktivitätsmuster für die drei Knoten der verdeckten Schicht gelernt. Diese entsprechen den Zuständen im endlichen Automat.
 - Unterschied zwischen Knoten im neuronalen Netz und Zuständen im Automat: Zustände entsprechen Aktivitätsvektoren.
 - Der Automat hat einen Zustand pro Zeitschritt. Das Netz hat einen Aktivitätsvektor in der verdeckten Schicht pro Zeitschritt.
- Ein rekurrentes Netz kann jeden endlichen Automat emulieren, ist aber exponentiell mächtiger. Mit N binären verdeckten Knoten können 2^N Aktivitätsvektoren codiert werden
 - Dies ist wichtig, wenn aus der Eingabe mehrere unabhängige Dinge repräsentiert werden müssen (Faktorisierung des Zustands).
 - Ein endlicher Automat ist hier nicht effizient.