COMPUTATIONAL VISION:

SEMANTIC IMAGE SEGMENTATION

Petia Radeva

COMPUTATIONAL VISION:

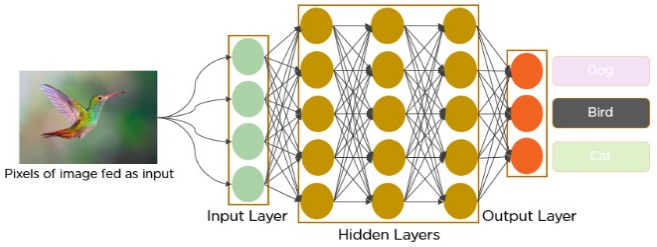
SEMANTIC IMAGE SEGMENTATION

Roser Sala-Llonch
roser.sala@ub.edu
Institute of Neurosciences
Department of Biomedicine
Faculty of Medicine

Index

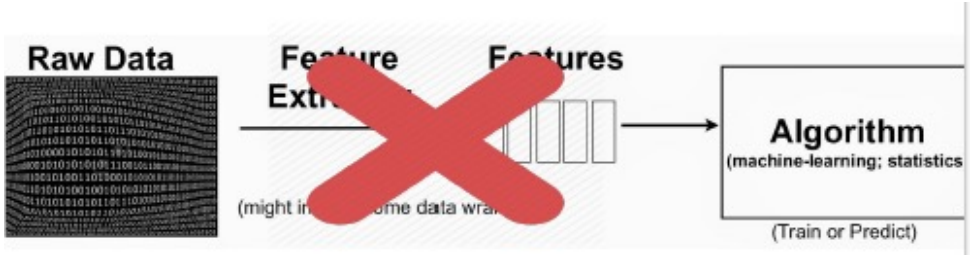
- Transfer learning
  - Taxonomy
  - Domain adaptation
  - Multi-task learning
  - Example: Food recognition
- Image segmentation
  - Semantic
  - Instance
  - Panoptic
  - Evaluation
- Conclusions

What makes DNN so popular?

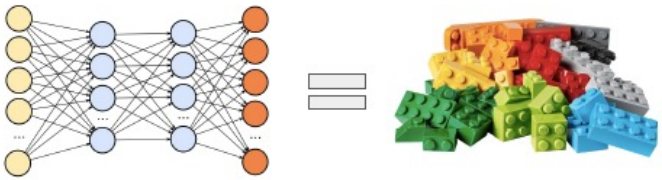


It has the three advantages:

1. Self-learned high-level features representations



2. Modularity



3. Transfer Learning (a.k.a. fine-tunning)

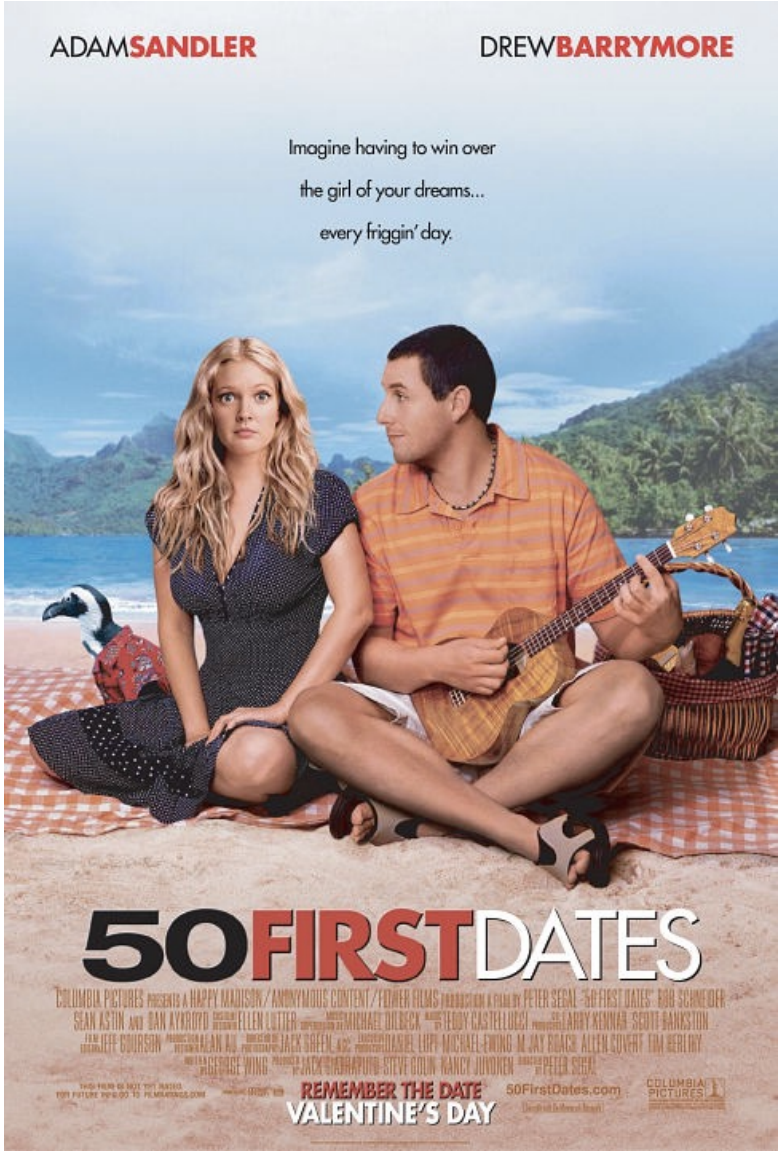
Let's imagine that…

Henry Roth is a man afraid of commitment up until he meets the beautiful Lucy.

They hit it off and Henry think he's finally found the girl of his dreams,

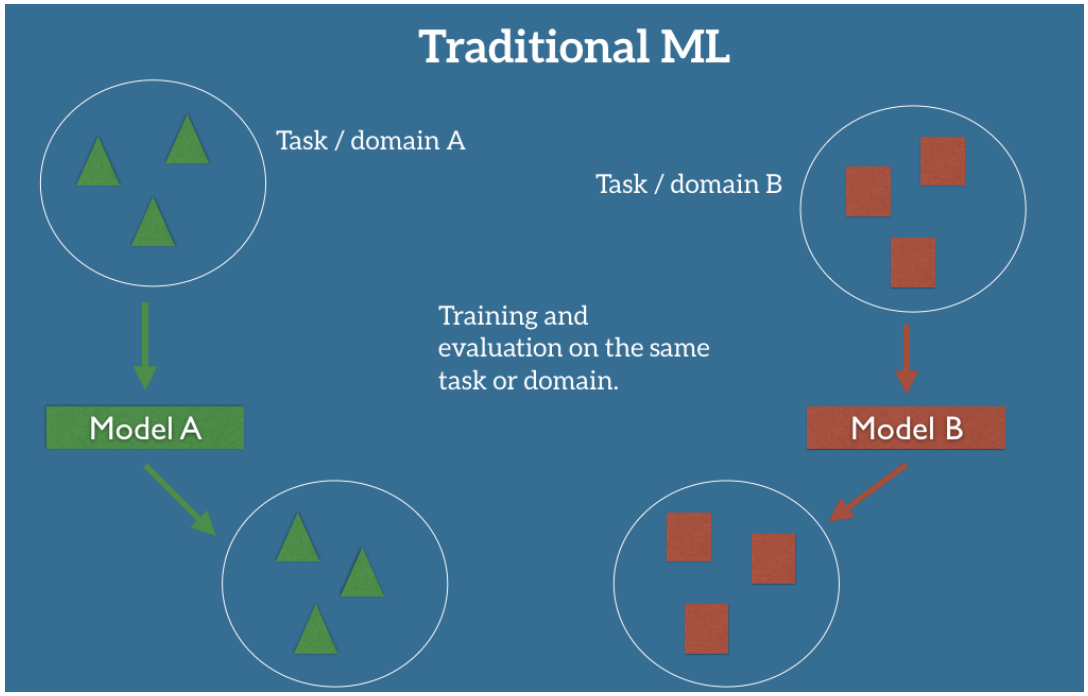
until he discovers:

she has short-term memory loss and forgets him the next day.



What is Transfer learning?

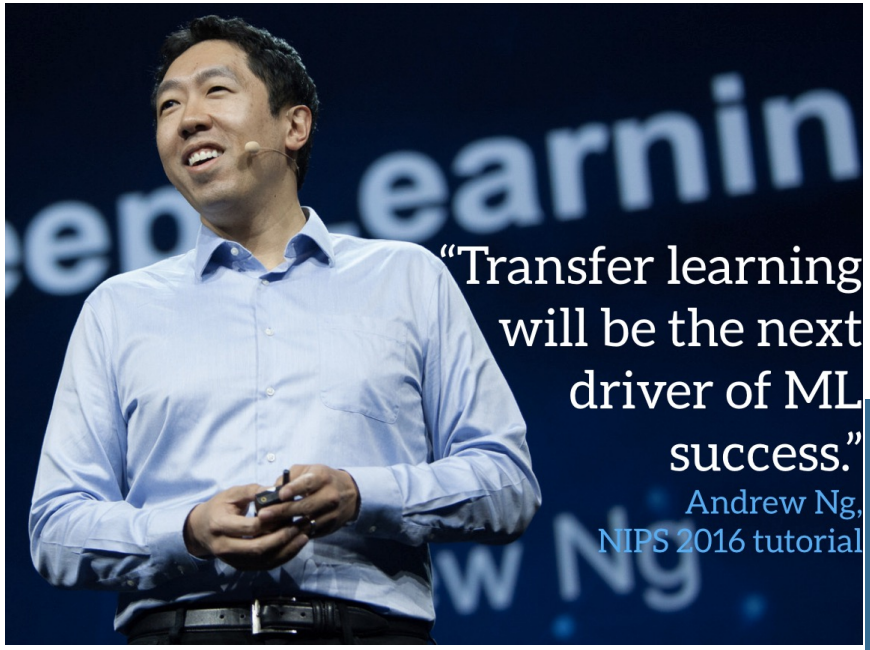
Transfer Learning (TL): The ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks (in new domains).

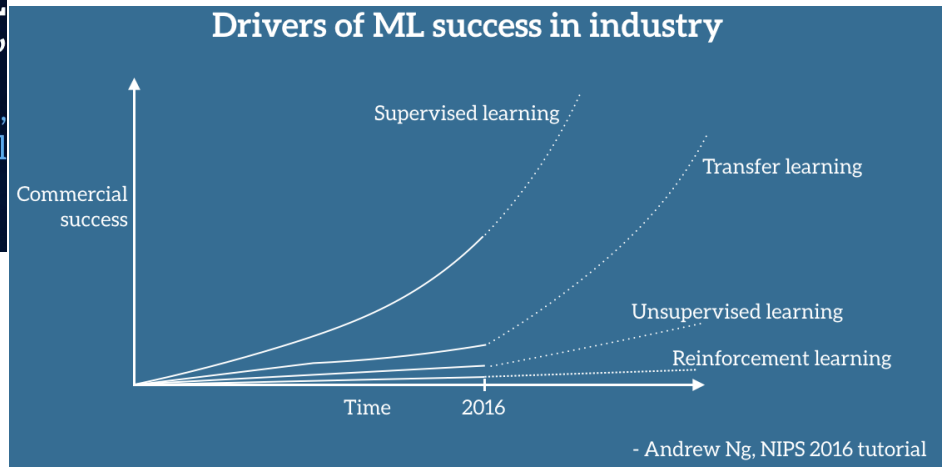


https://ruder.io/transfer-learning/index.html#whatistransferlearning

Why Transfer Learning Now?

Andrew Ng, chief scientist at Baidu and professor at Stanford

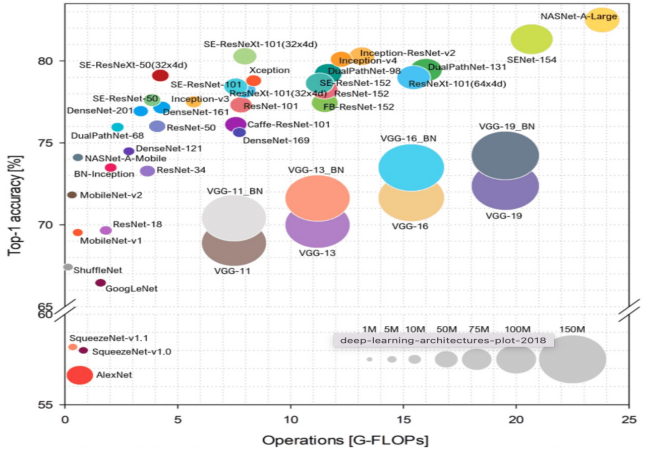




https://ruder.io/transfer-learning/index.html#fn44

Why now is the time for Transfer Learning?

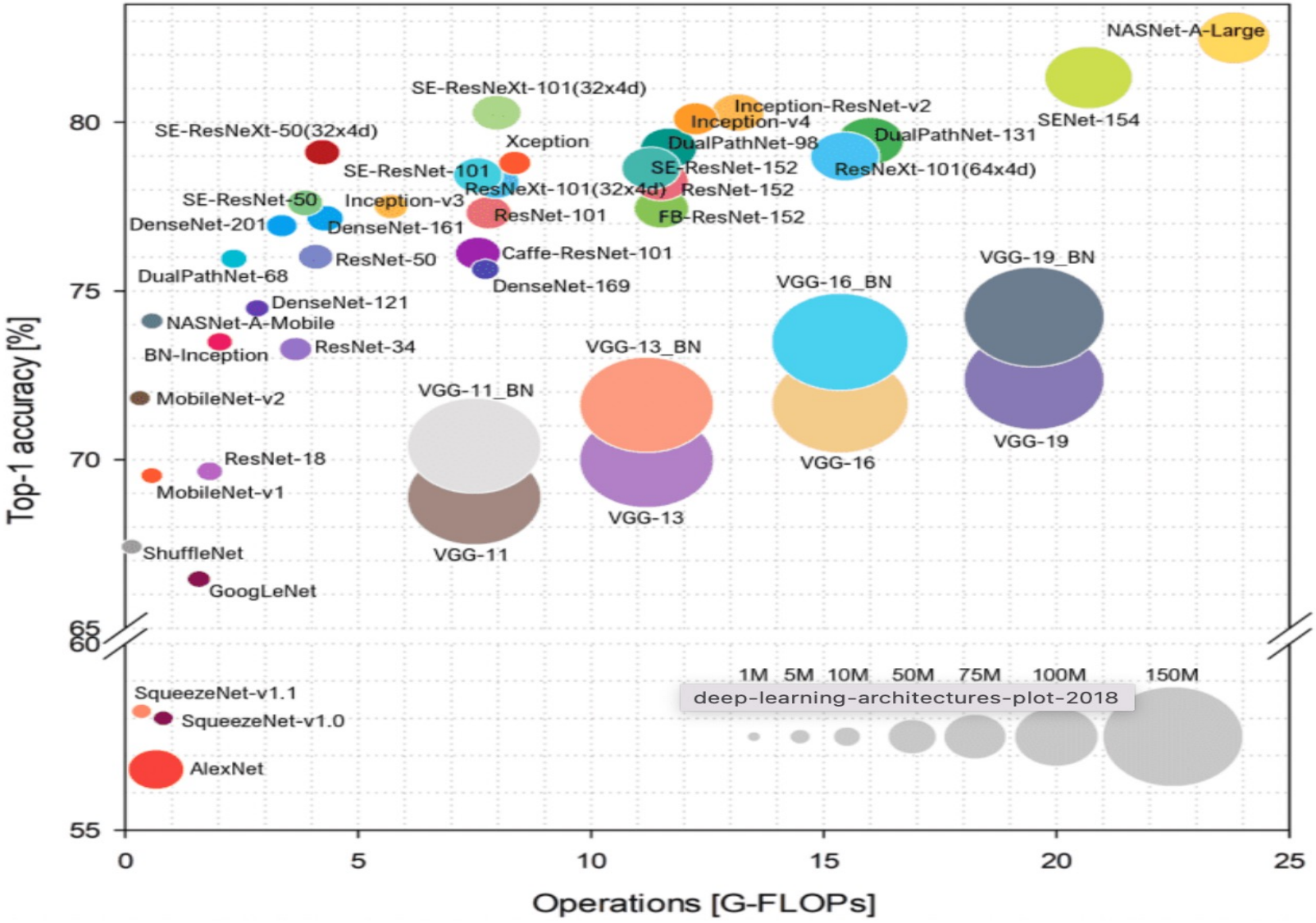
Availability of well-working models



- ResNet achieving **superhuman performance** on recognizing objects
- Google's Smart Reply automatically handles **10% of all mobile re**sponses;
- speech recognition error has consistently dropped and is **more accurate than typing**;
- we can automatically identify **skin cancer as well as dermat**ologists;
- **Google's NMT system** is used in production for more than 10 language pairs, etc.

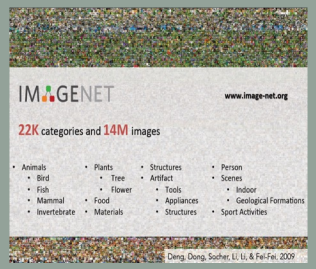
https://ruder.io/transfer-learning/index.html#fn44

Why now is the time for Transfer Learning?

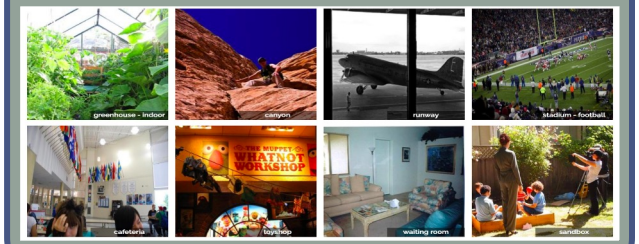


Why now is the time for Transfer Learning?

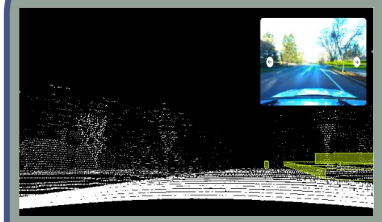
- Successful models are immensely **data-hungry** and rely on **huge amounts of labeled data** to achieve their performance.
- A model is asked to behave in a **new situation on tasks** different from what it was trained for.



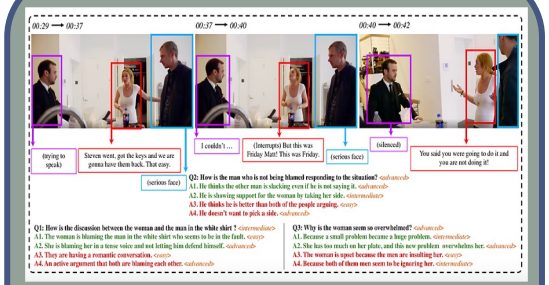
LVIS Challenge: 2.2M masks, 16K images



Places2: 10M images



Lyft Level 5



SocialIQ

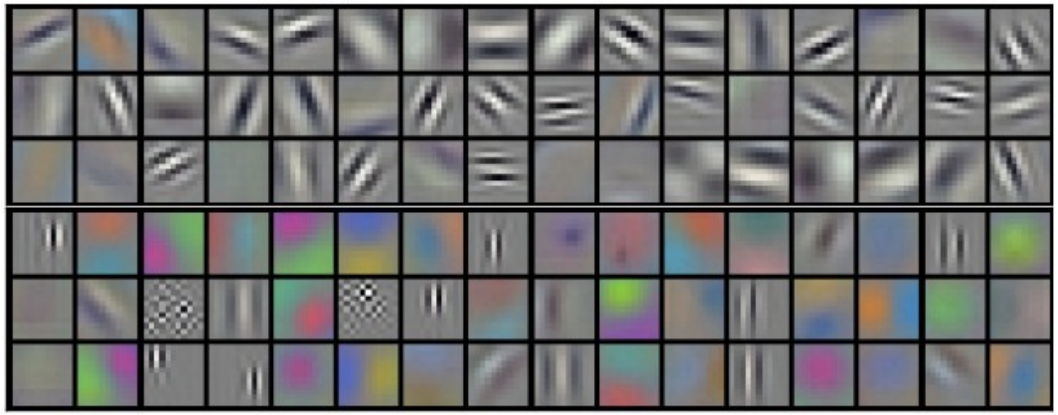
TACO: Waste in the wild

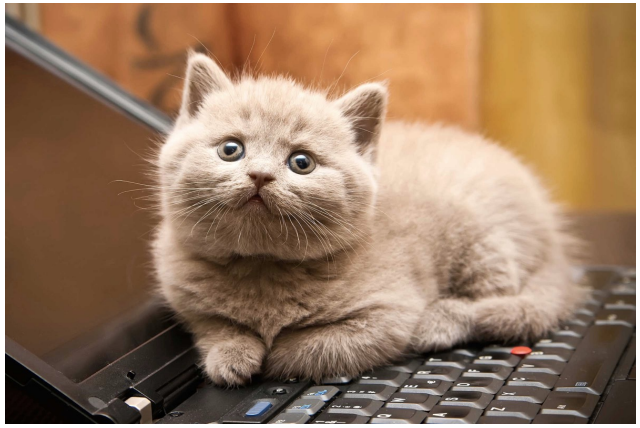
FastMRI

https://ruder.io/transfer-learning/index.html#fn44

Transfer Learning Methods: Using pre-trained CNN features

Understanding CNNs:

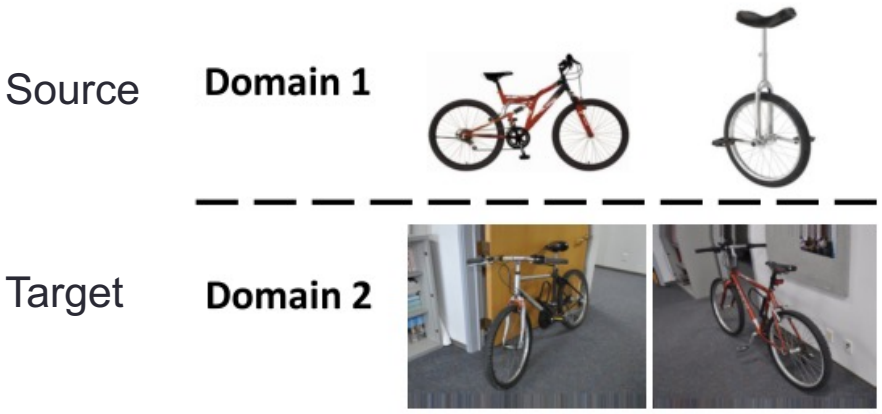


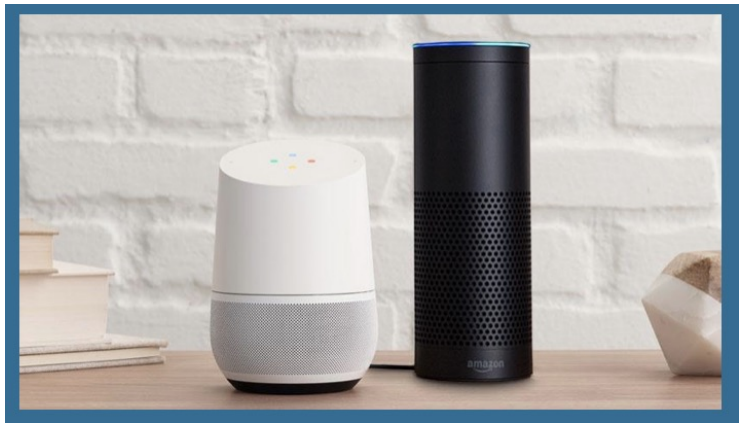


- Lower convolutional layers capture low-level image features, e.g. edges, while higher convolutional layers capture more and more complex details, such as body parts, faces, and other compositional features.
- Classification is done by FC layers that explain how edges and shapes are combined.
- For a new task, use the off-the-shelf features of a state-of-the-art CNN pre-trained on ImageNet and train a new model on these extracted features

Fine-tuning: Adapting to new domains

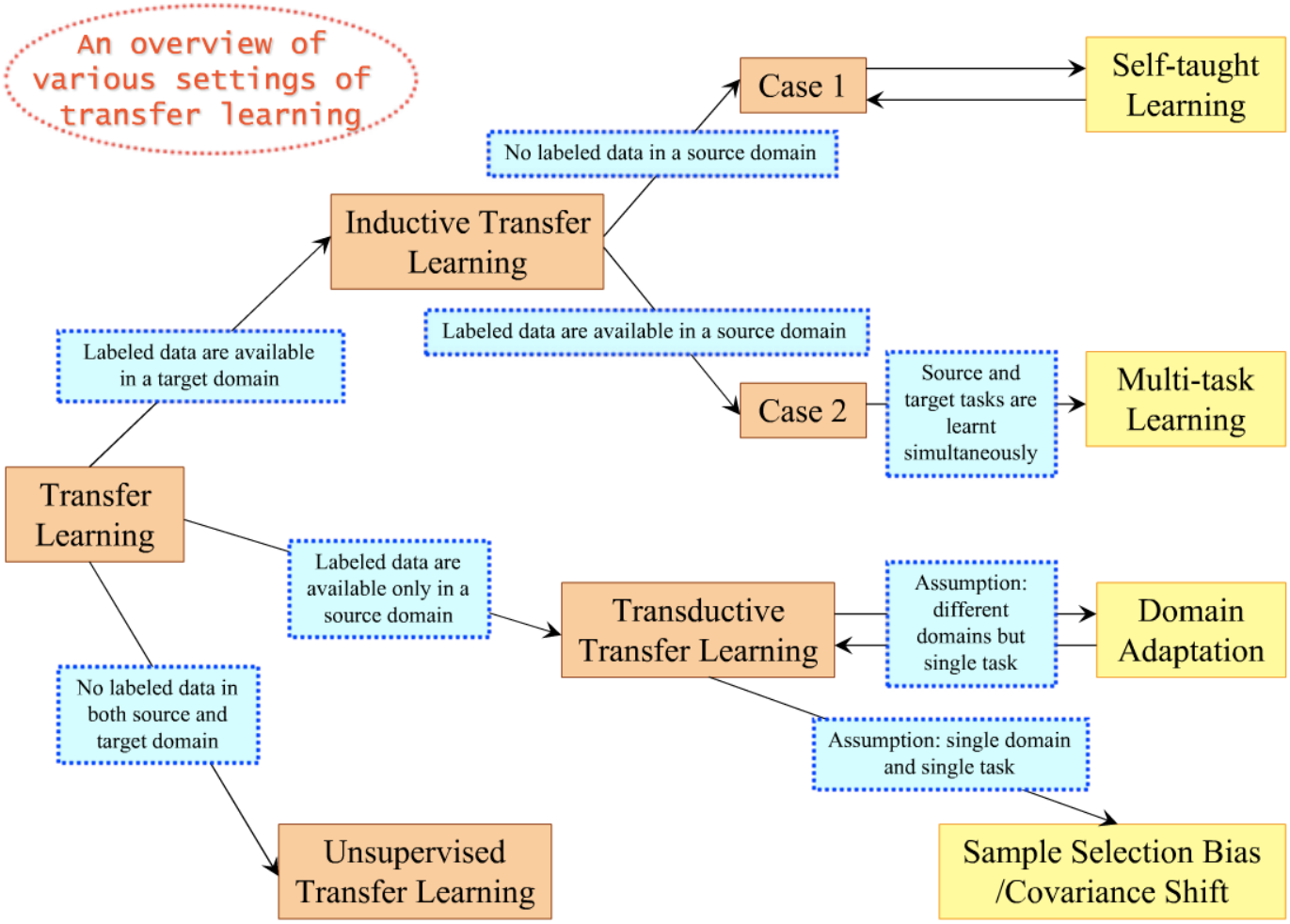
- Often the data where **labeled information is easily accessible**
  - However, the training data could contain a **bias** imperceptible to humans
    - even if the training and the test data look the same,
- **The goal**: to easy the training in a new domain.





- Continual learning:
  - avoid forgetting

Taxonomy of Transfer Learning

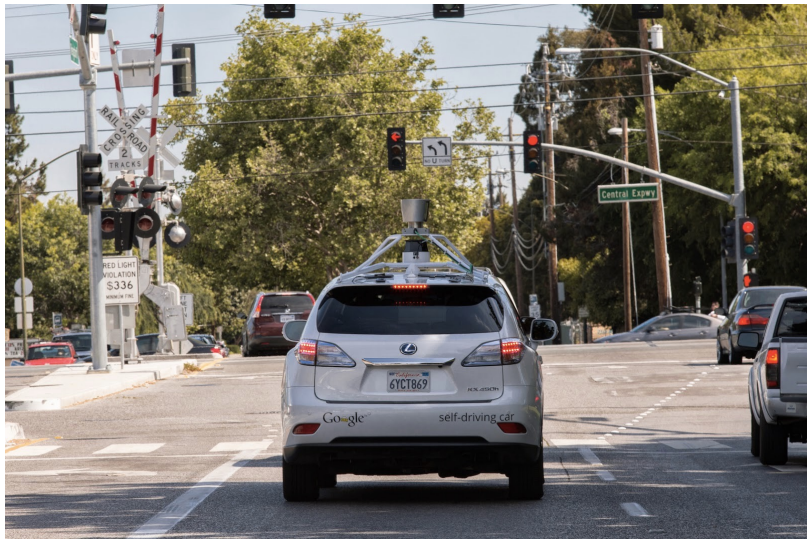


Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Trans. KDE, 22(10), 1345-1359

Domain adaptation: Learning from simulations

- Gathering data and training a model in the real world is either **expensive, time-consuming, or simply too dangerous**.

- Learning from a simulation and applying the acquired knowledge to the real world is an instance of transfer learning,
  - as the feature spaces between source and target domain are the same,
  - but the marginal probability distributions between simulation and reality are different,
  - this difference diminishes as simulations get more realistic.



- Learning from synthetic data: application in medical imaging.

Multi-Task Learning

- Input: A lot of labeled data in a source domain and the target domain that coincide.
- Goal: Classifying the target and source domain data simultaneously.
- Assumption: The source domain and target domain data share some common features, which can help classifying sharing knowledge.
- Main Idea: To train simultaneously both tasks.

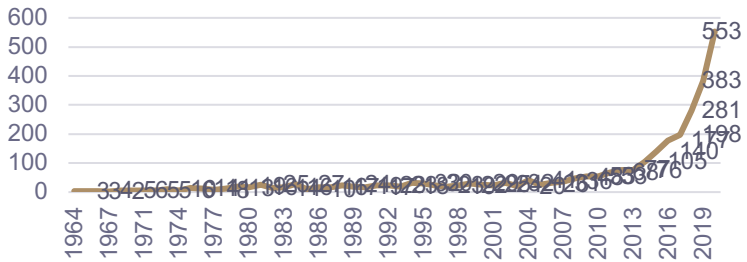
Let us consider the problem of food recognition

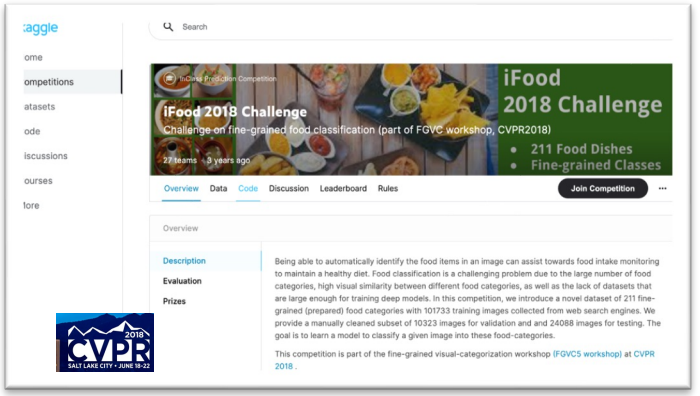
Did you know that:

- **180 million photos** with the hashtag #food on Instagram
- **90 new photos** hash-tagged #foodporn are uploaded to Instagram **every minute**.
- **54%** of 18–24 year olds take a food photo while eating out.
- **39%** have posted it somewhere online.
- **5% of over-50s** share food snaps on forums as Facebook & Twitter

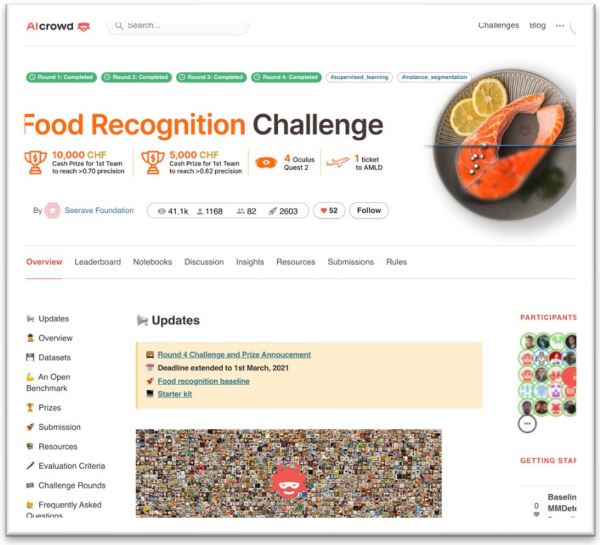
"Camera eats first"

Food recognition popularity

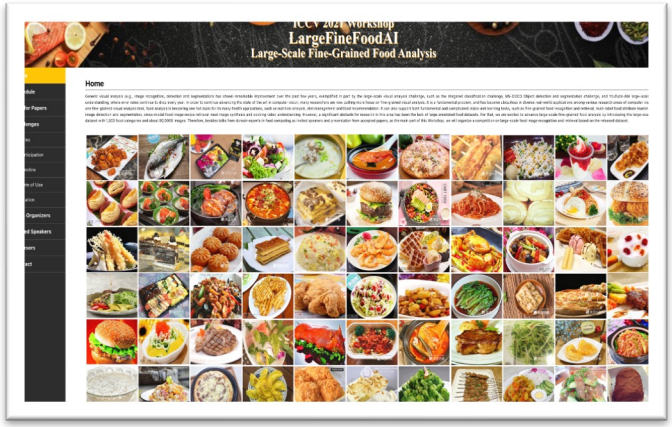




iFood 2011 fine-grained (prepared) food categories with 135.733 images



AICrowd: 26000 annotated segmented images



LargeFineFoodAI: 1,000 fine-grained food categories and over 50,000 images.

Why is the food recognition a challenge?

Difficulties

Huge intra-class variations

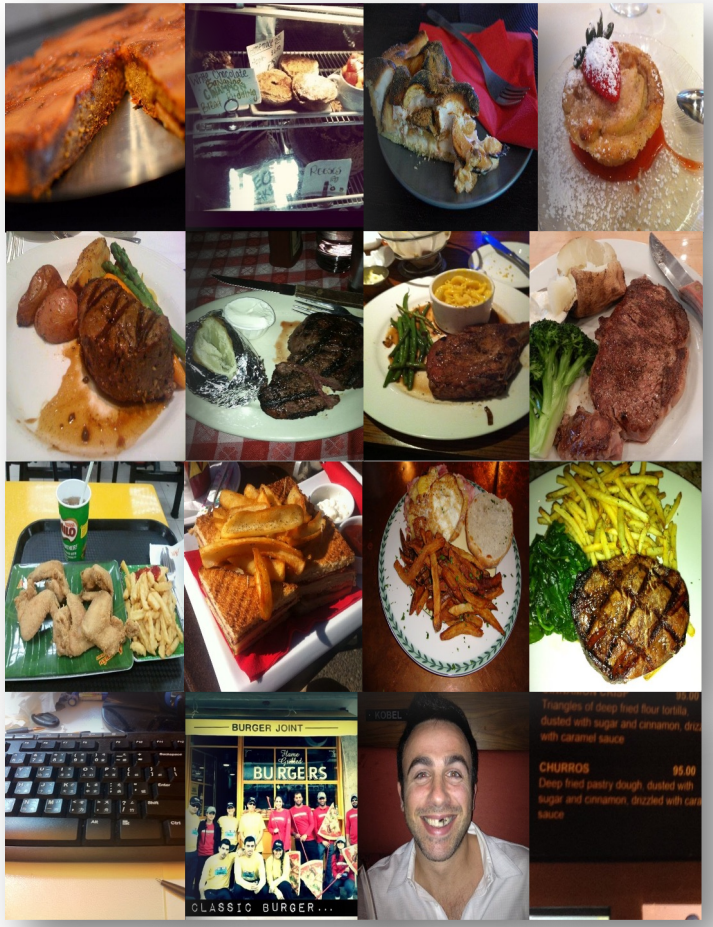
Ambiguous definition

Inter-class similarities

Mixed items

Need of huge datasets

Bad Labeled



What to do when you have a really complicate problem?

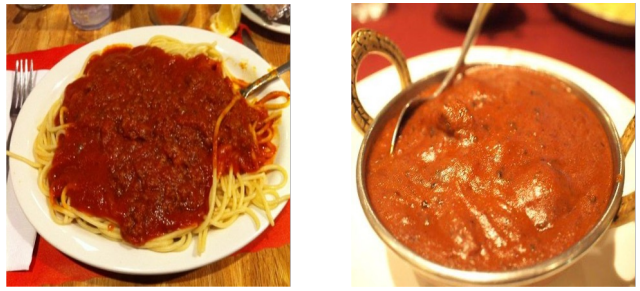
Motivation

Food Analysis Problems

Ingredients

- Intra-class variability
- Inter-class similarity

Intra-class variability example: Apple. Image source: Recipes5k



Inter-class similarity example: Tomato sauce and Curry sauce. Image source: Recipes5k

Decreasement in Precision

The food recognition is a Fine-grained recognition problem

Food datasets

Food256: 25.600 images (100 images/class) Classes: 256

Food101 – 101.000 images
(1000 images/class)
Classes: 101

FoodX-251
Classes: 251
140K images

Food1K
Classes: 1000
370K images

Food DB	ImageNet	Future Food DB
150.000 images 231 categories	1.400.000 images 1000 categories	????? images 200.000 categories

Are we able to recognize thousands of dishes?

- **79.46%** - best performance on the food dataset UECFOOD-100
  - **72%** in the extended version **UECFOOD-256** [1]

- **44.1% -** baseline performance in the largest dataset **ChinFood1000 -** [2]

  - It is well-known that the class number is inversely proportional to the model performance [3]

  - How to achieve scalability?

[1] Martinel, Niki and Foresti, Gian Luca and Micheloni, Christian, Wide-slice residual networks for food recognition, IEEE WACV, 2018.
[2] Fu, Zhihui and Chen, Dan and Li, Hongyu, ChinFood1000: a large benchmark dataset for Chinese food recognition, ICIC 2017.
[3] Luo, C., Li, X., Yin, J., He, J., Gao, D., Zhou, J.: How does the data set and the number of categories affect cnn-based image classification performance? Journal of Software, 2019.

Multi-Task Learning (MTL)

- Learning **multiple objectives** from a shared representation
  - *Efficiency* and prediction *accuracy*

Cuisine: French.

Categories: Meat.

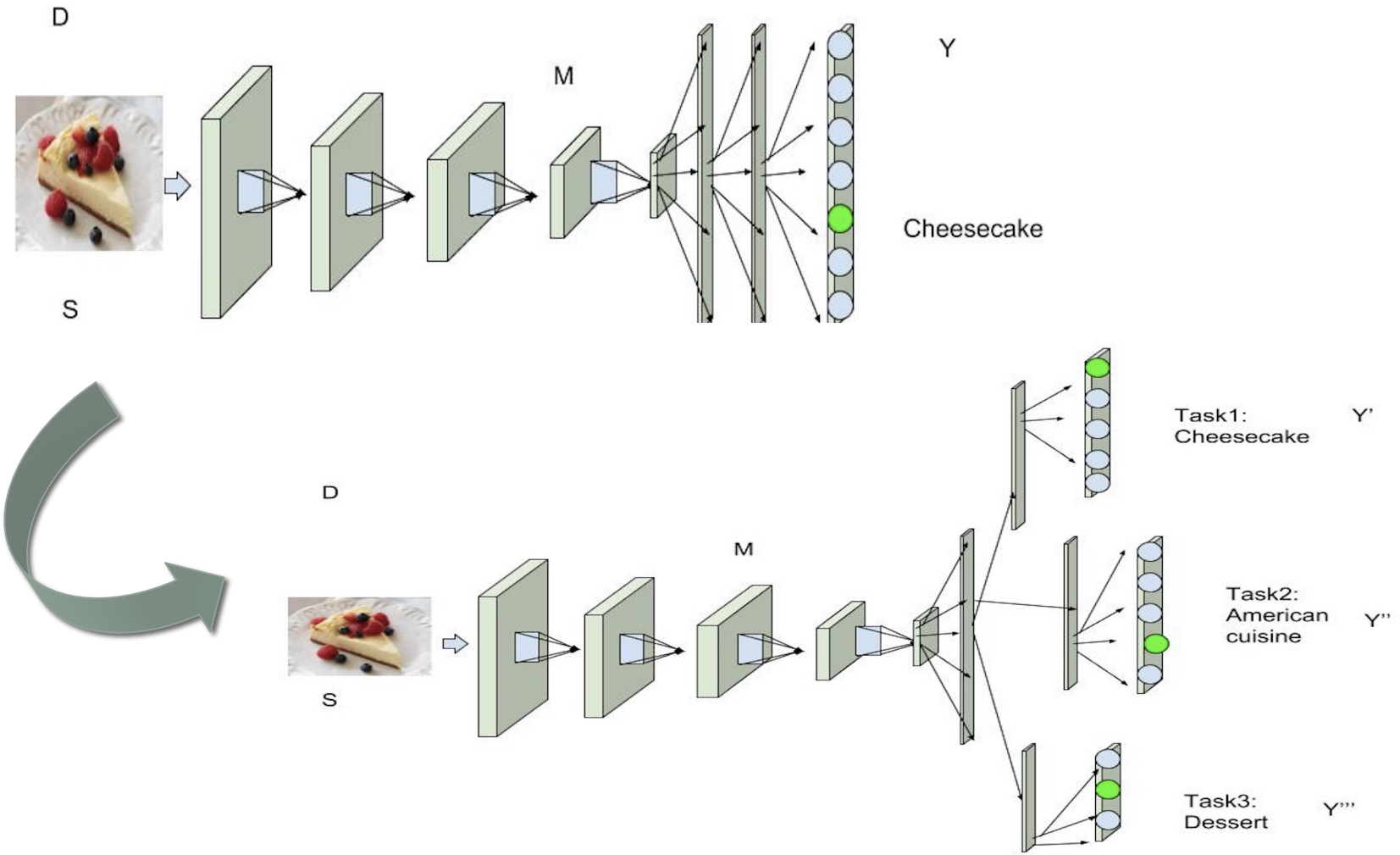
Ingredients: salt, oil, onion, garlic, black pepper, tomato, cloves, parsley, thyme, bay, white wine, clove, duck, fat, mutton.

Dish: Confit de canard.

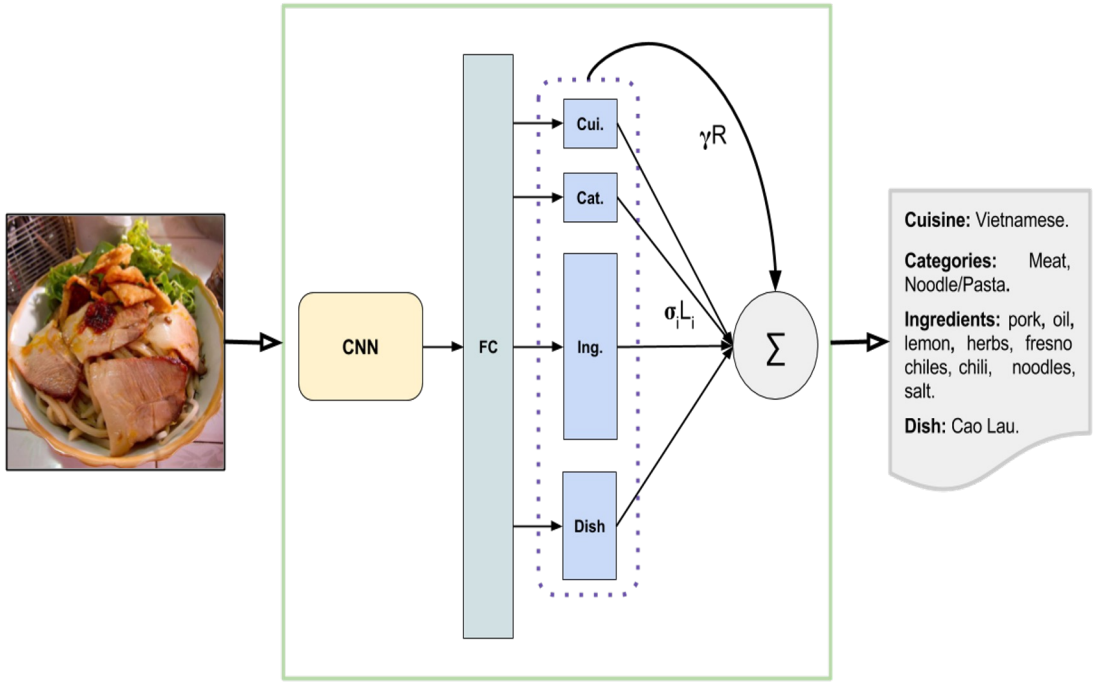
- Crucial importance in systems where **long computation** run-time is prohibitive
  - Combining all tasks *reduces computation*.

- Inductive **knowledge transfer**
  - *Generalization* by sharing the domain information between complimentary tasks.

Transfer Learning



Food Recognition as a MTL



$$L_{total} = \sum_i \omega_i L_i$$

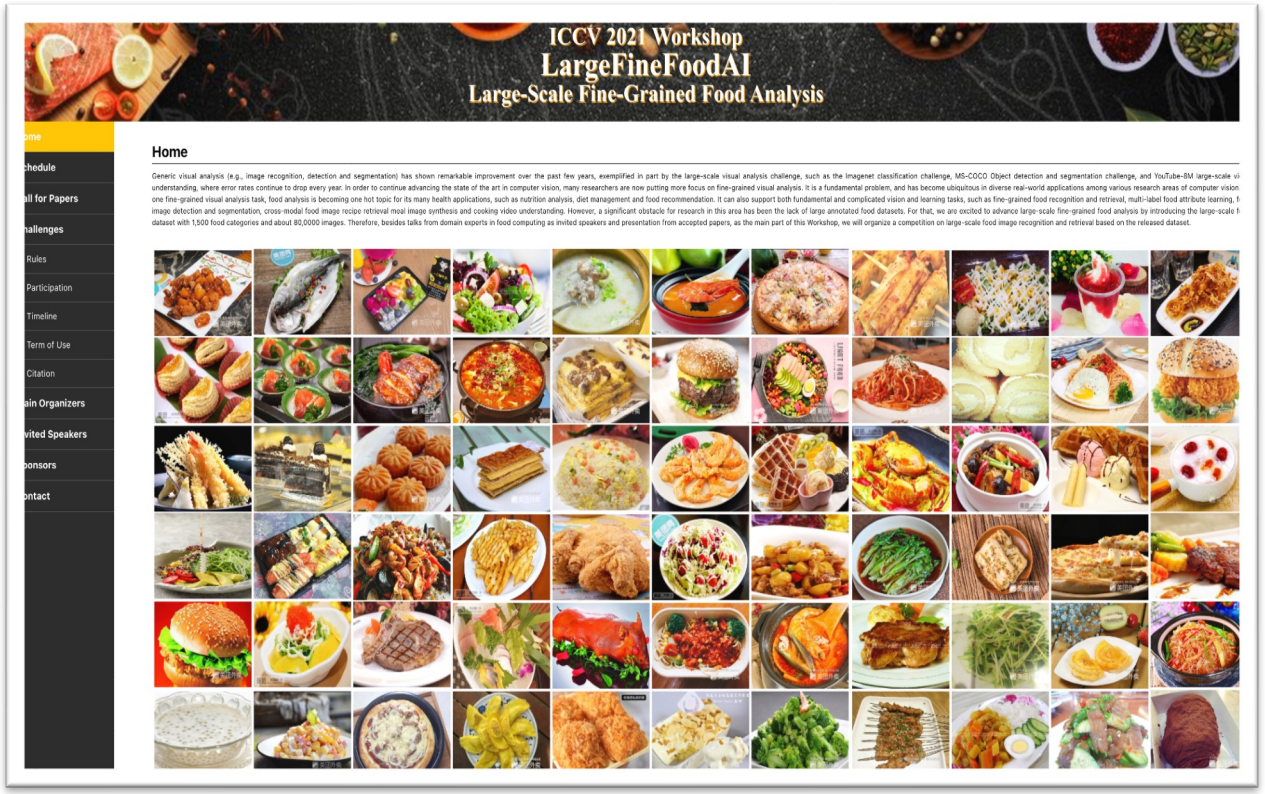
How the determine the importance (weigths) of each class?

How to define the importance of each task?

- Weighted uniformly the losses.
- Manually tuned the losses.
- Dynamic weighted of the losses.
  - The main task is fixed and weights are learned for each side-task ([1]).
  - Weight the tasks according to the homoscedastic uncertainty ([2]).

[1] X. Yin and X. Liu. Multi-task convolutional neural network for face recognition.
[2] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics.

Food datasets



LargeFineFoodAI dataset 1,000 fine-grained food categories and over 50,000 images.

MAFood Dataset

- Food – 450 dishes, 11 categories, 11 cuisines
- Ingredients – 65
- Drinks – 40

In total:
more than
550.000 images

19:45

Results

	GT	RUMTL	Single-task
	**Dish:** tacos	**Dish:** tacos	**Dish:** prime_rib
	Cuisine: mexican	**Cuisine:** mexican	**Cuisine:** american
	Categories: vegetable, meat, bread	**Categories:** vegetable, bread	**Categories:** vegetable, meat

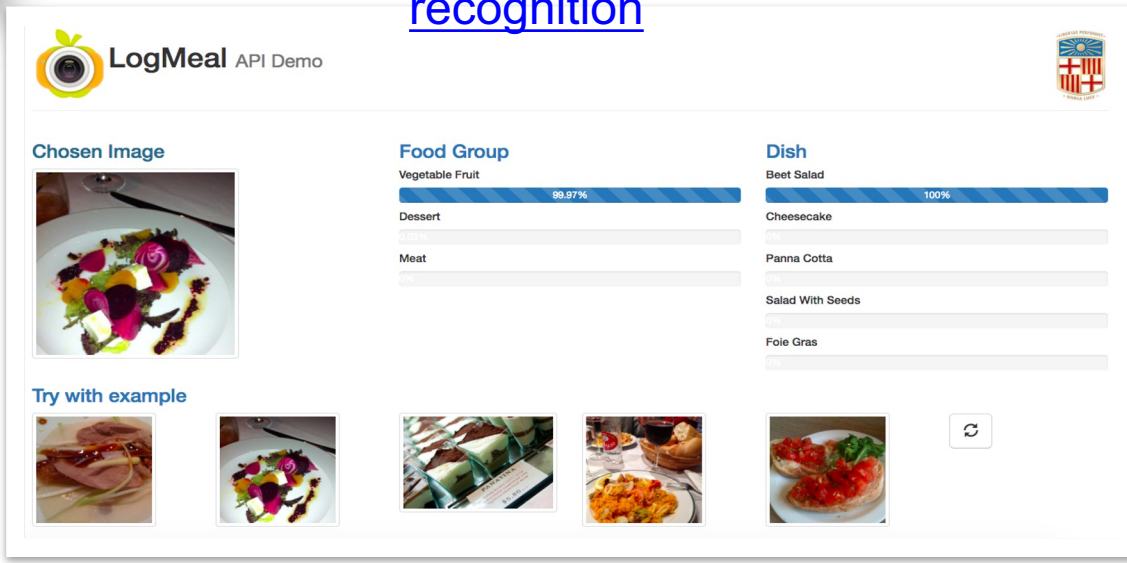
	GT	RUMTL	Single-task
	**Dish:** eggs_benedict	**Dish:** eggs_benedict	**Dish:** ravioli
	Cuisine: american	**Cuisine:** american	**Cuisine:** italian
	Categories: vegetable, bread, egg	**Categories:** vegetable, bread, egg	**Categories:** vegetable, egg

	GT	RUMTL	Single-task
	**Dish:** sushi	**Dish:** sushi	**Dish:** cha_ca
	Cuisine: japanese	**Cuisine:** japanese	**Cuisine:** japanese
	Categories: vegetable, seafood, rice	**Categories:** seafood, rice	**Categories:** fried_food

	GT	RUMTL	Single-task
	**Dish:** ravioli	**Dish:** bruschetta	**Dish:** lobster_roll_sandwich
	Cuisine: italian	**Cuisine:** italian	**Cuisine:** italian
	Categories: dumpling	**Categories:** vegetable, bread	**Categories:** vegetable, meat, bread

Food ingredients recognition

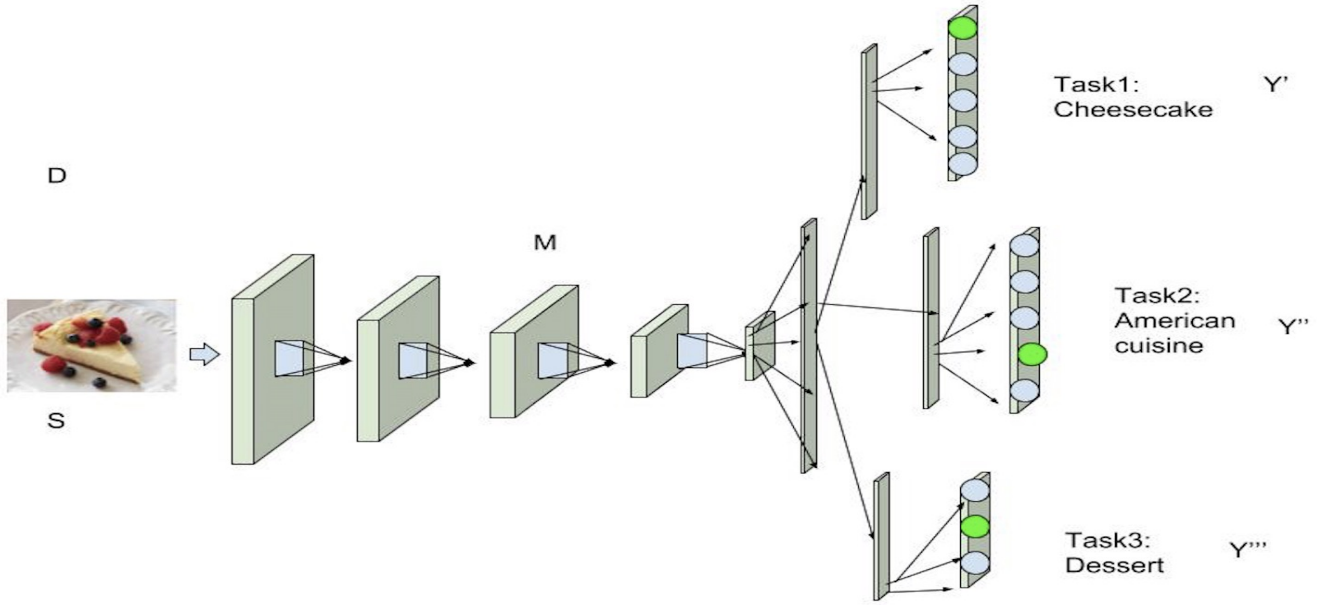
[Food category and class recognition]



Understanding the cooking process

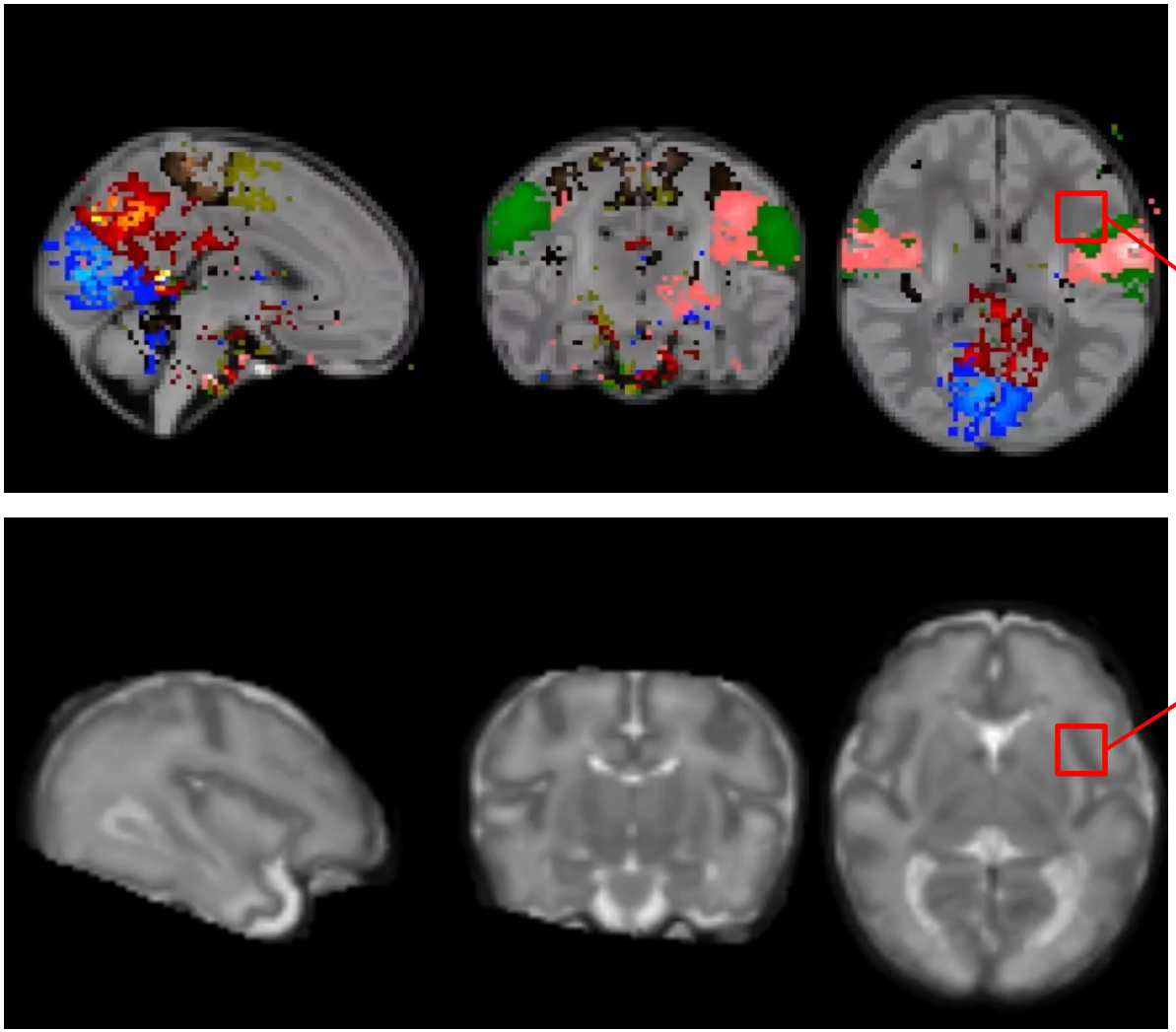
Epic-kitchen:

Can you give an example of MTL from our previous lectures?



Shell all tasks be of same type (e.g. classification)?

MTL & Neuroimaging



2 different modalities.
Learned Together

Index

- Transfer learning
  - Taxonomy
  - Domain adaptation
  - Multi-task learning
  - Example: Food recognition

  - Image segmentation
  - Semantic
  - Instance
  - Panoptic
  - Evaluation

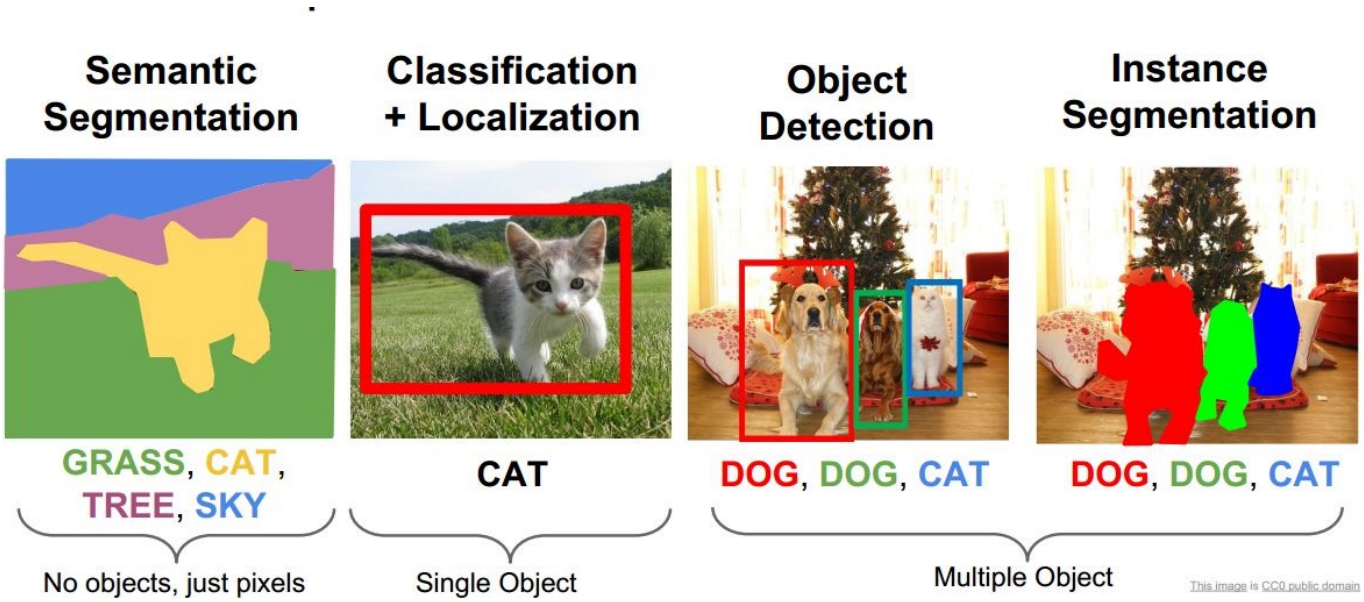
- Conclusions

CVPR'2022

	MENU ▾ Select Primary Subject Area	# papers	Oral	Poster
1	Recognition: detection, categorization, retrieval	177	25	152
2	Image and video synthesis and generation	157	26	131
3	3D from multi-view and sensors	137	26	111
4	Low-level vision	110	19	91
5	Vision + language	105	20	85
6	Segmentation, grouping and shape analysis	99	16	83
7	Transfer/low-shot/long-tail learning	86	15	71
8	Deep learning architectures and techniques	85	20	65
9	Self-& semi-& meta- & unsupervised learning	84	7	77
10	Video analysis and understanding	77	15	62
11	Pose estimation and tracking	62	14	48
12	Representation learning	61	11	50
13	3D from single images	60	10	50
14	Scene analysis and understanding	56	9	47
15	Face and gestures	54	7	47
16	Computational photography	53	10	43
17	Motion and tracking	53	8	45
18	Adversarial attack and defense	52	10	42
19	Datasets and evaluation	52	7	45
20	Machine learning	41	7	34
21	Action and event recognition	40	8	32
22	Efficient learning and inferences	40	3	37
23	Medical, biological and cell microscopy	37	5	32
24	Vision applications and systems	32	4	28
25	Navigation and autonomous driving	31	2	29
26	Vision + graphics	24	6	18
27	Privacy and federated learning	21	3	18
28	Vision + X	20	4	16
29	Physics-based vision and shape-from-X	16	2	14
30	Robot vision	16	3	
31	Explainable computer vision	15	2	13
32	Demo	15		15
33	Optimization methods	14	2	12
34	Transparency, fairness, accountability, privacy and ..	14	4	
35	Document analysis and understanding	12		12
36	Biometrics	11	2	9

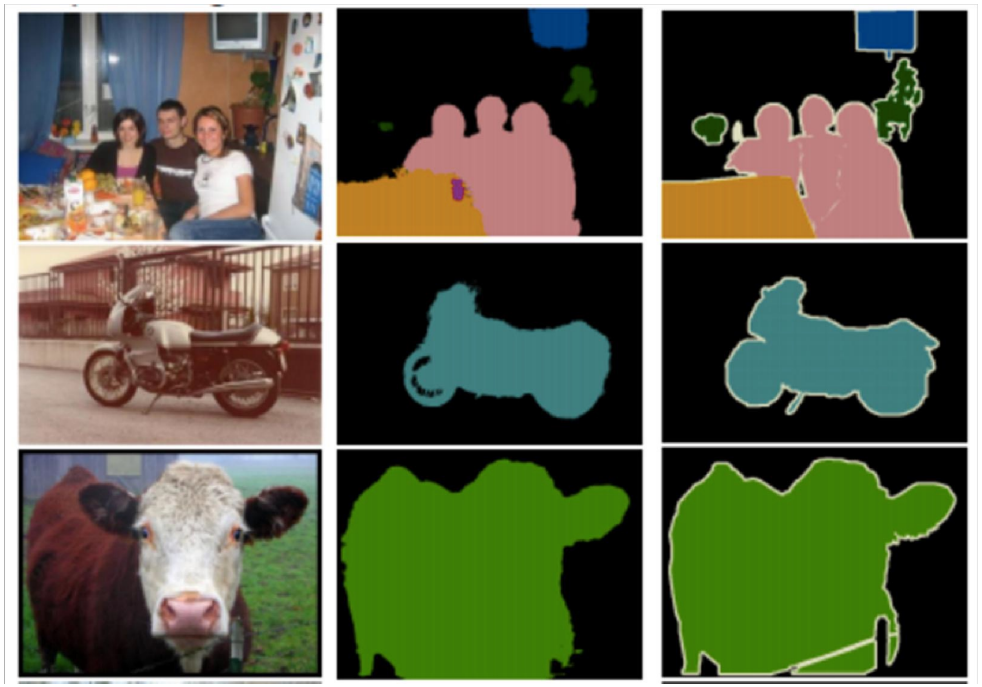
Categories at CVPR 2022 ranked by number of papers accepted

Semantic Segmentation

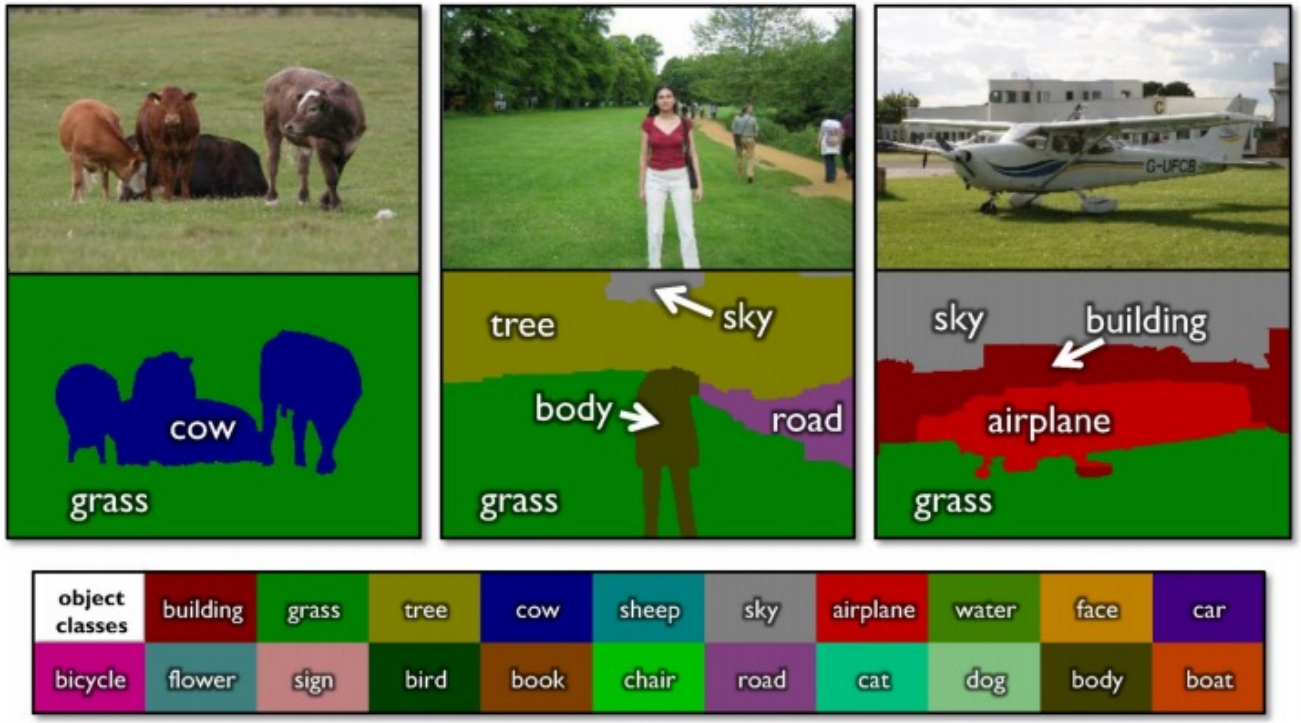


Semantic Segmentation

- Classifying each pixel in the image
- Recognizing and delineating objects in an image



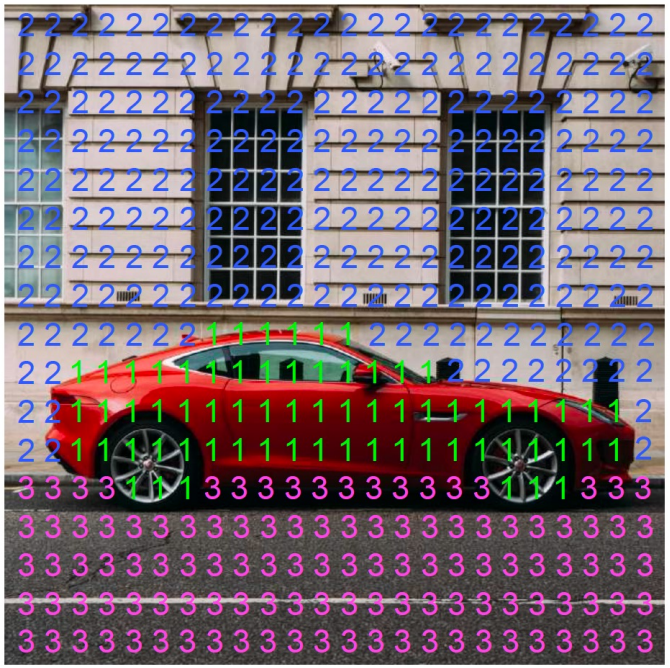
Pixel-wise Semantic Segmentation



- Label every pixel!
- Don’t differentiate instances
- Classic computer vision problem

http://cs231n.stanford.edu/slides/2016/winter1516_lecture13.pdf

Pixel-wise Semantic Segmentation

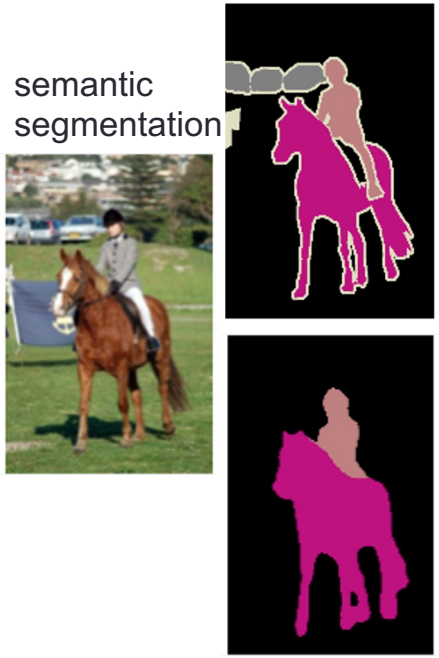


1 – Car
2 – Building
3 – Road

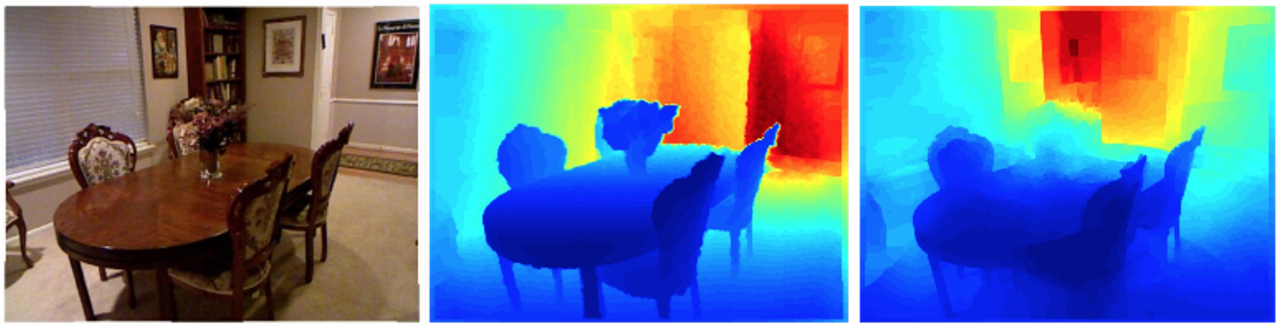
A segmentation map

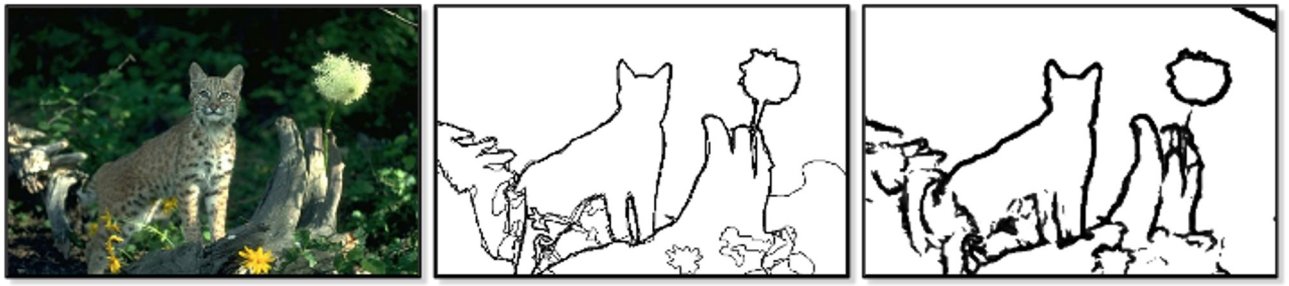
Pixels in, pixels out

semantic segmentation



monocular depth estimation (Liu et al. 2015)





boundary prediction (Xie & Tu 2015)

Image segmentation can be defined in different ways

Slide credit: Jonathan Long

Why Semantic Segmentation?

- Road scenes understanding
- Useful for autonomous navigation of cars and drones

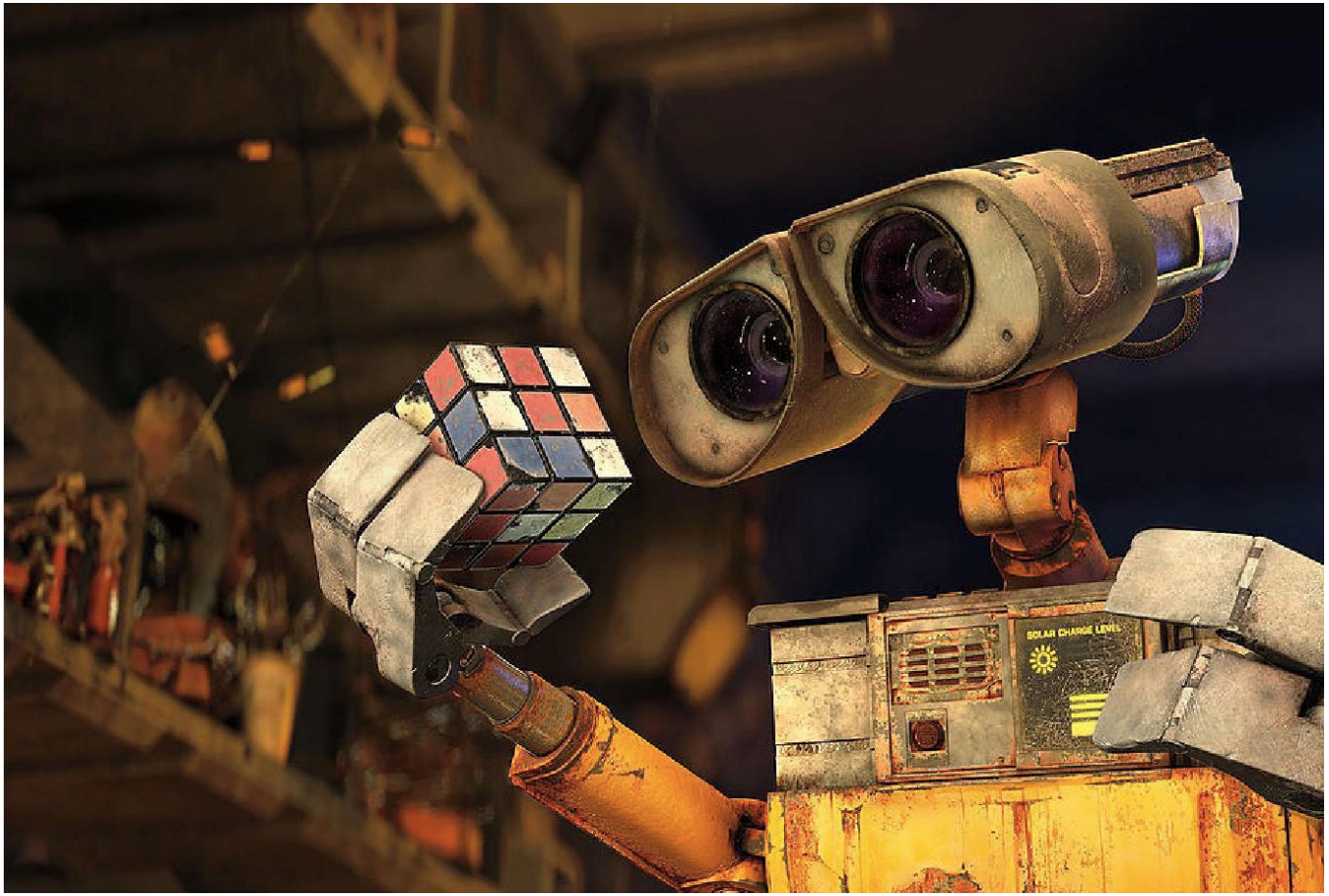
Image taken from the cityscapes dataset.

Why Semantic Segmentation?

- To help partially sighted people by highlighting important objects in their glasses

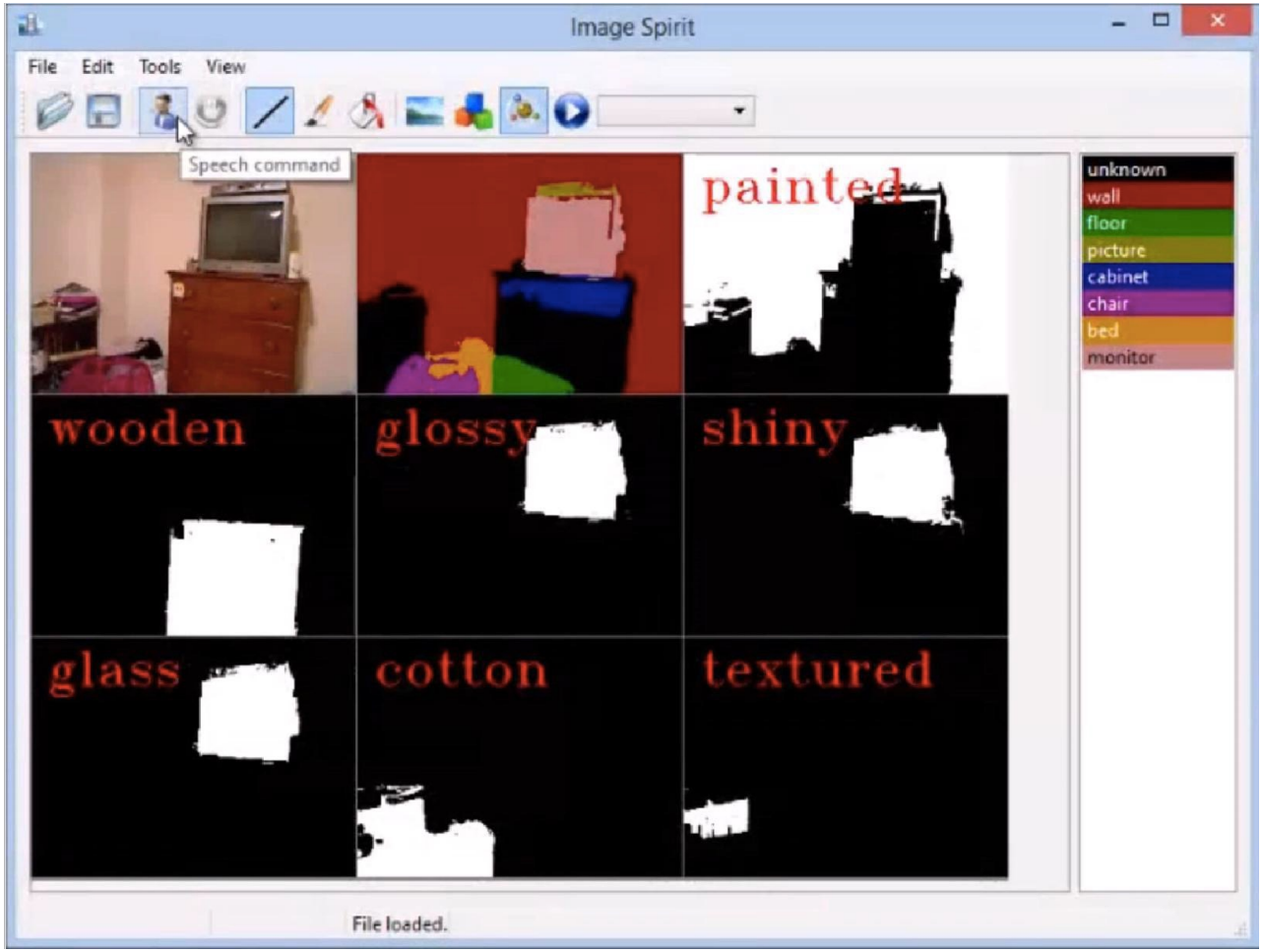
Why Semantic Segmentation?

- To let robots segment objects so that they can grasp them



Why Semantic Segmentation?

- Useful tool for editing images



Why Semantic Segmentation?

- Medical purposes: e.g. segmenting
- tumours, dental cavities, ...

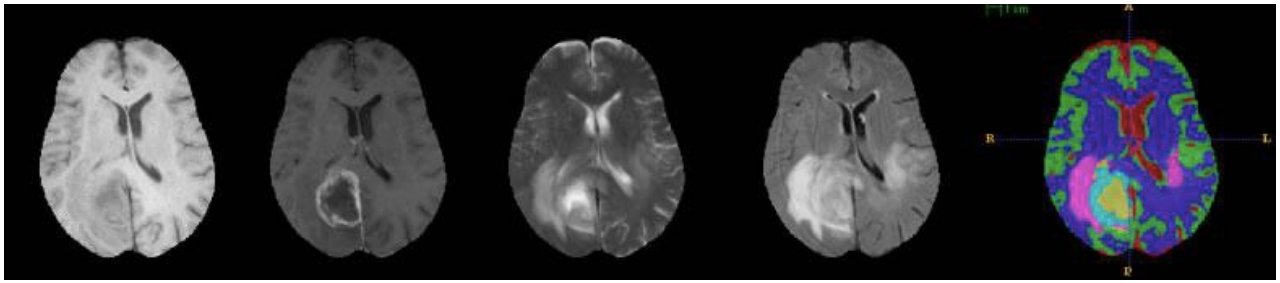
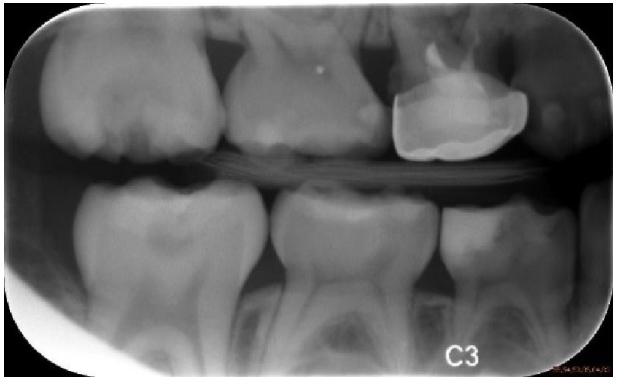
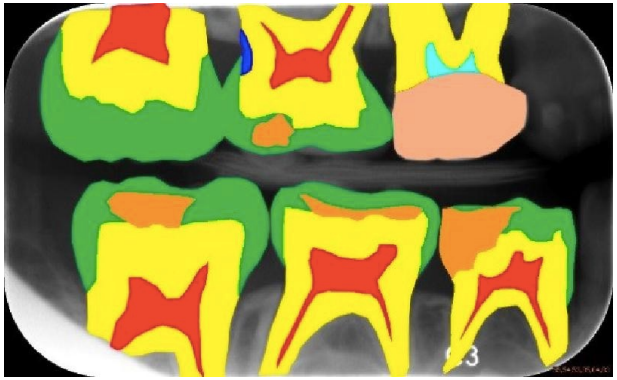


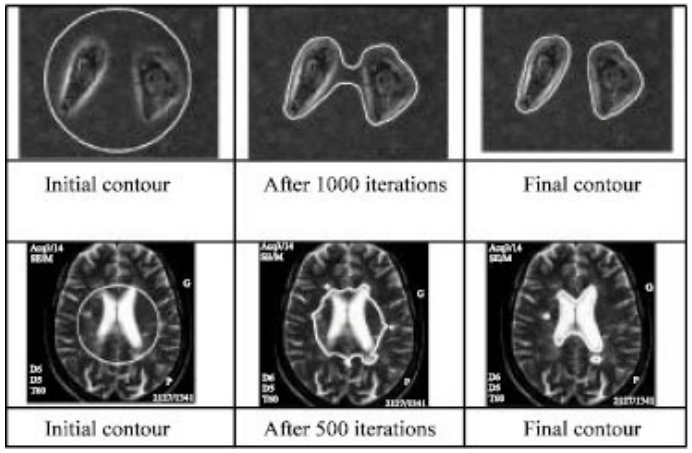
Image taken from Mauricio Reyes

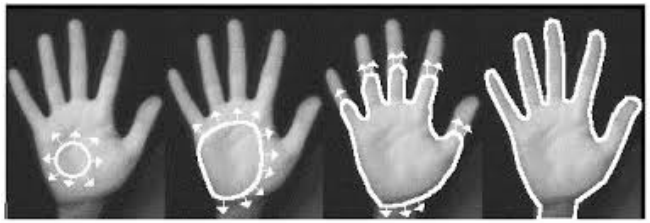




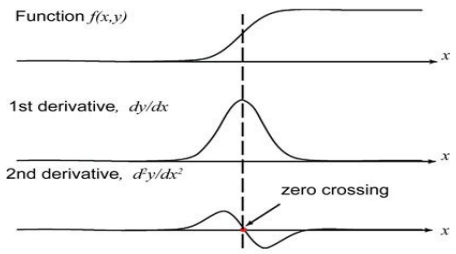
ISBI Challenge 2015, dental x-ray images

Classical Segmentation methods

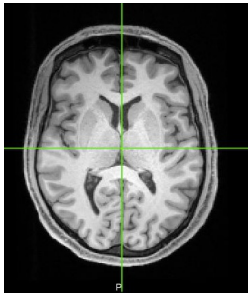


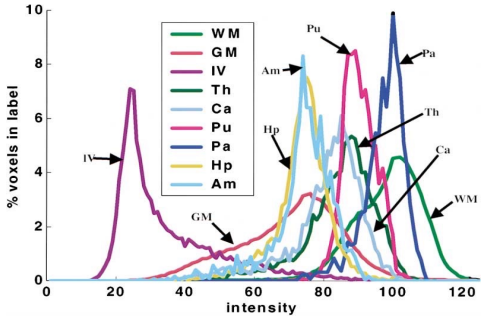


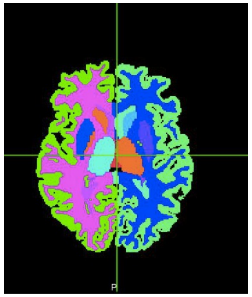
Optimization steps towards an energy minimum of an **active contour** algorithm applied to medical images. From Debakla et al. (2011)



Canny edge detector
Representation of pixel intensity *y* as a function of *x, f(x)* and its first and second order derivatives along a border







Histogram-based methods
Fishl et al., 2000

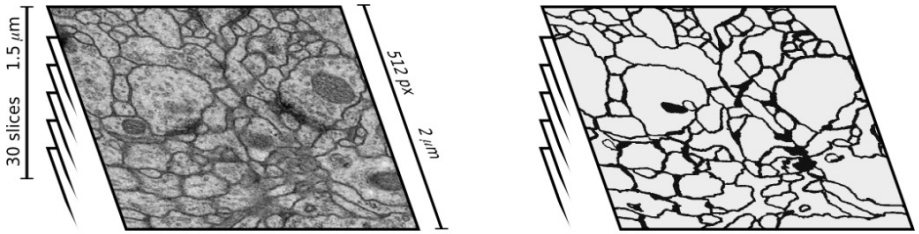
Object segmentation today is based on Deep learning

He, Gkioxari, Dollár, Girshick. Mask R-CNN. In ICCV 2017

Biomedical Image Segmentation

- Thousands of labeled training images for image segmentation are usually beyond reach
  - 1 million training images

- The desired output should include **localization**



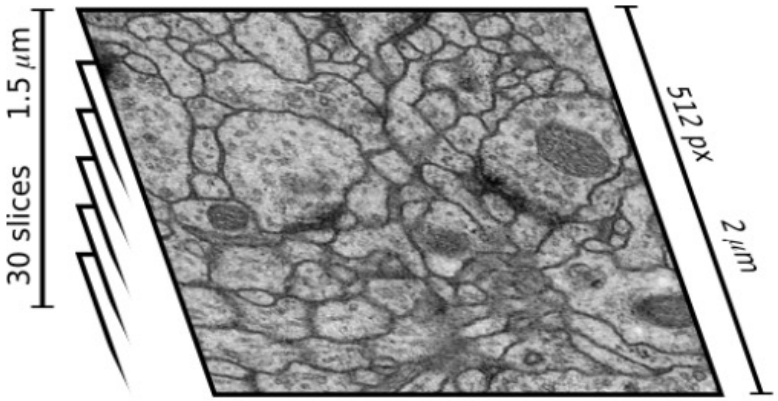
https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

ISBI 2012 Task

Predict the class label of each pixel

- Stacks of Electron microscopy (EM) images
- EM segmentation challenge at ISBI 2012
- 30 training images

Black - neuron membranes
White - cells

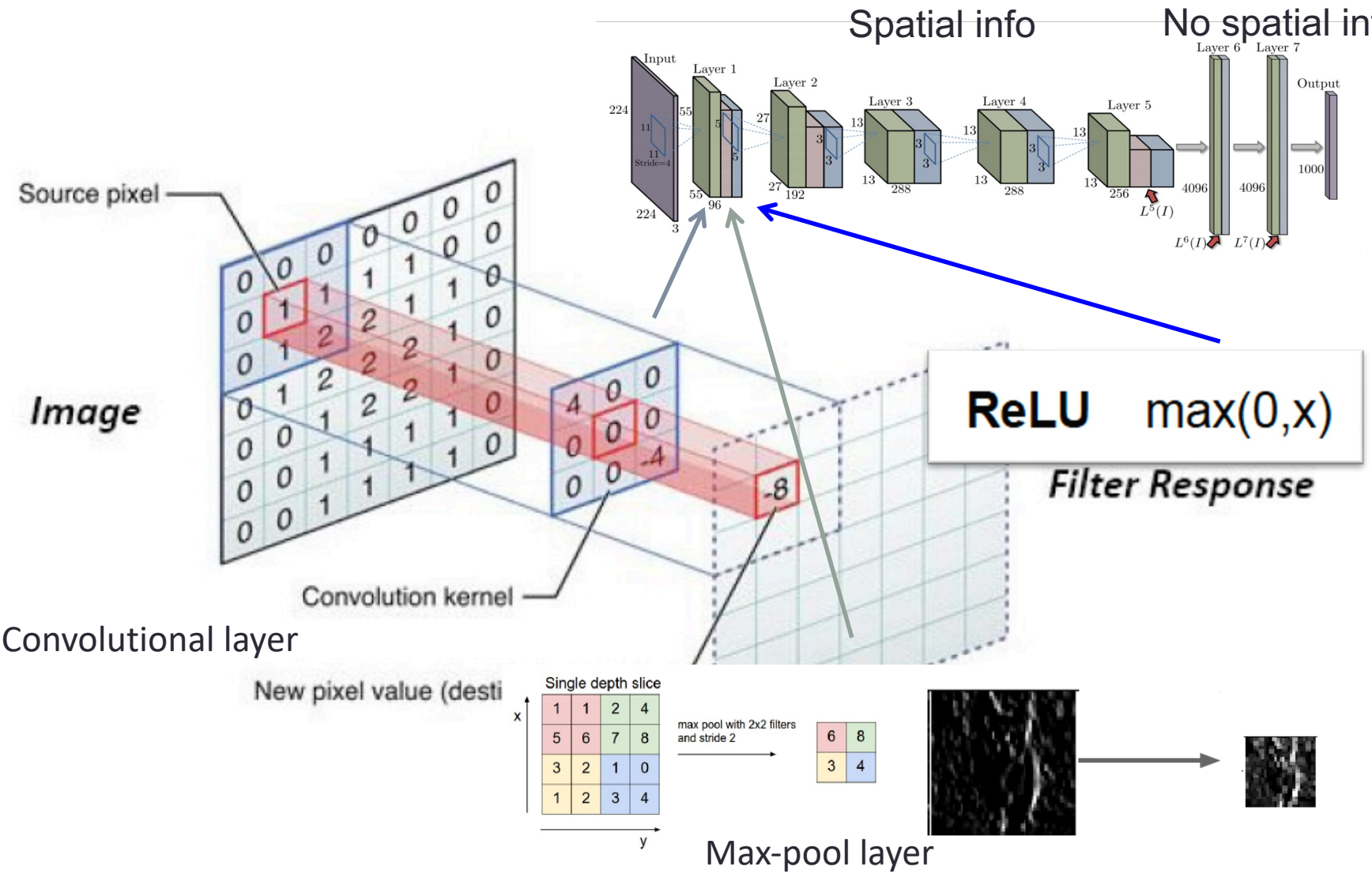


Training stack

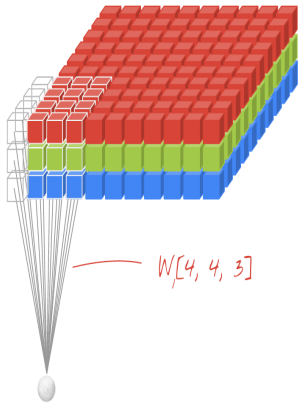
Ground truth

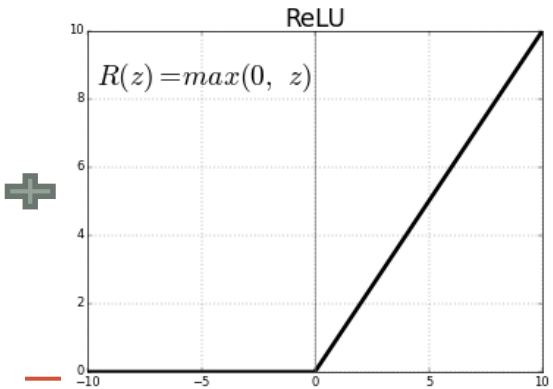
Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural net- works segment neuronal membranes in electron microscopy images. In: NIPS. pp. 2852{2860 (2012)

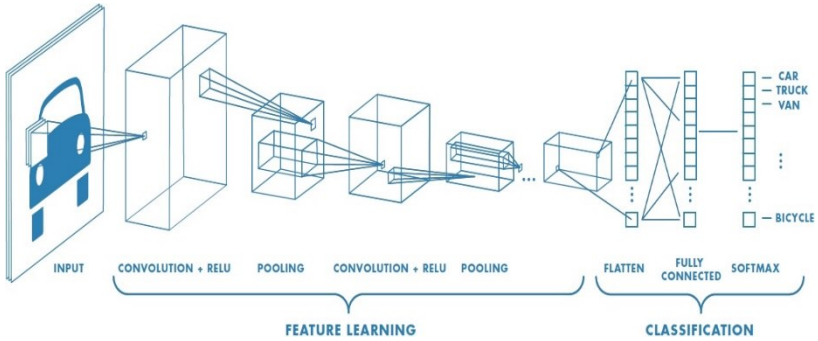
Remember: Convolutional Neural Network layers



CNNs







- Padding = 0
- Strides = 1

Output Size = (5−3+2∗0)/1+1 = 3

kernel

$$\begin{pmatrix} 0 & 1 & 2 \\ 2 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix}$$

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

12	12	17
10	17	19
9	6	14

W - Input volume size

F – Receptive field size (Filter Size)

P - Zero padding used on the border

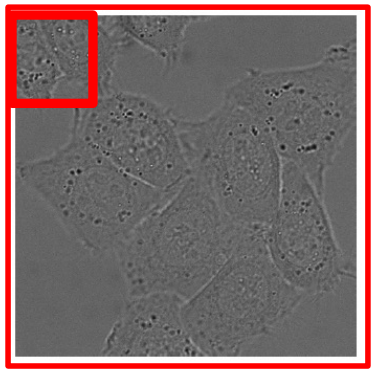
S - Stride

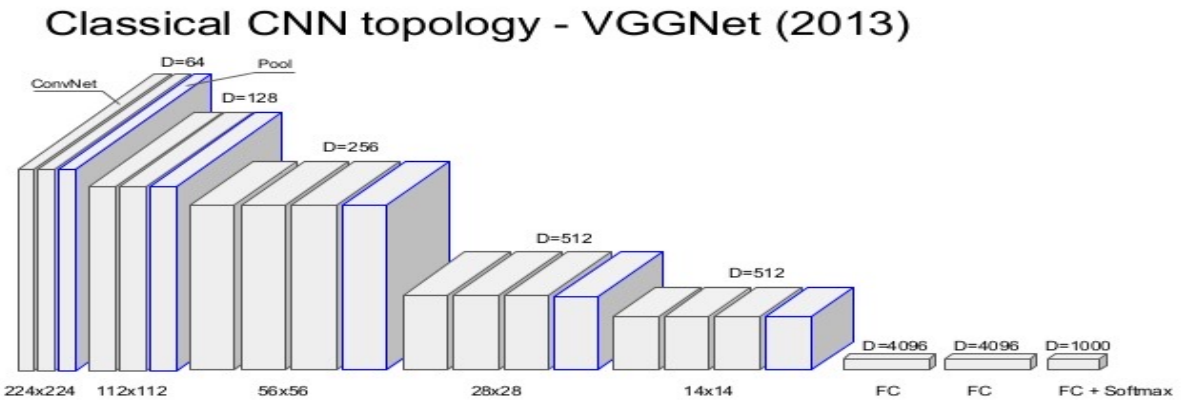
Output Size = (W−F+2P)/S+1

http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html
https://sites.google.com/site/nttrungmtwiki/home/it/data-science---python/tensorflow/tensorflow-and-deep-learning-part-3?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1

Challenge:
Segmentation of Neuronal Structures in EM

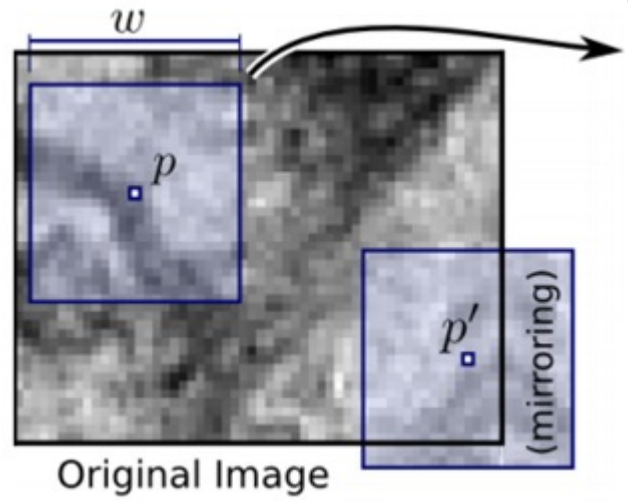
Remember the sliding window approach?



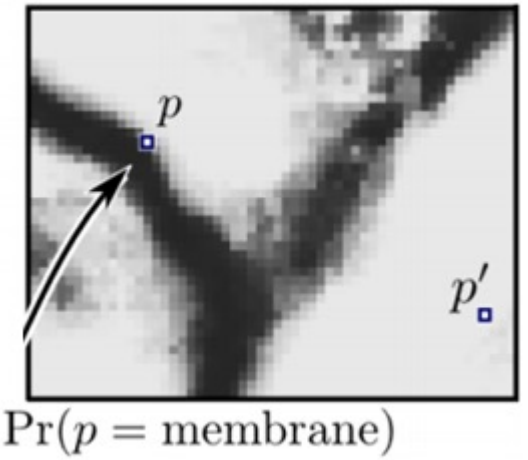


The DNN winner (ISBI 2012)

- Trained a network in a sliding-window (local region (patch) around that pixel)
  - ✓ **This network can localize**
  - ✓ **The training data in terms of patches is much larger than the number of training images**
  - x **Slow because the network must be run separately for each patch**
  - x **There is a lot of redundancy**



Deep Neural Netwok



Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural net- works segment neuronal membranes in electron microscopy images.

The winner (ISBI 2012)

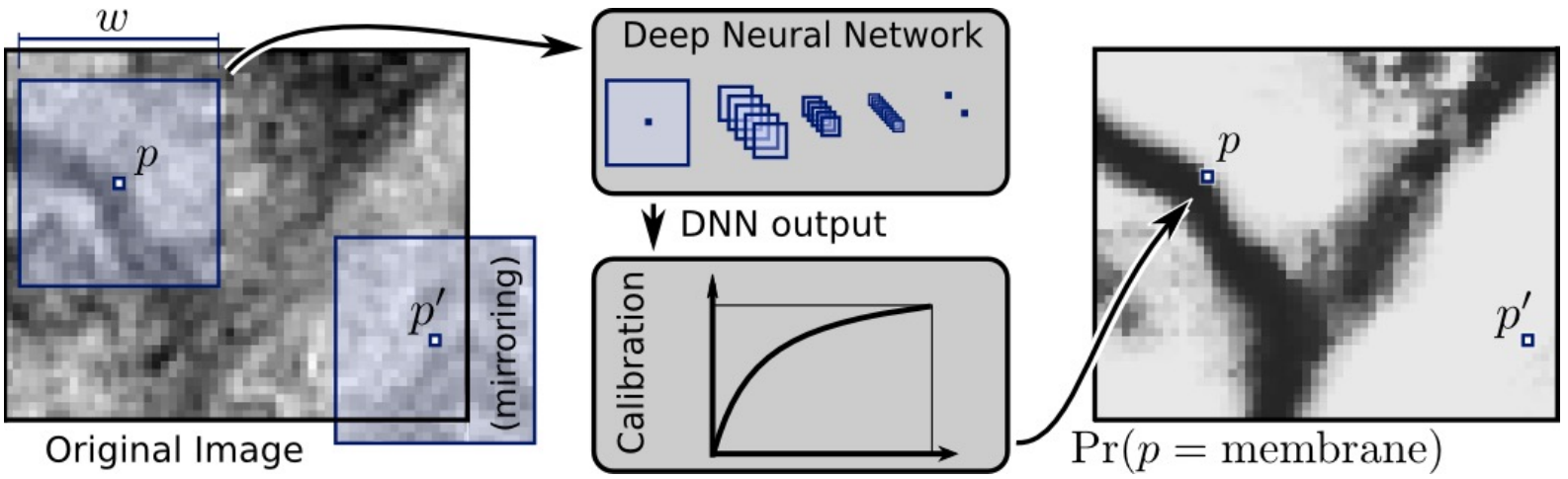
- Trade-off between localization accuracy and the use of context.

Larger patches: Require more max-pooling layers → reduce the localization accuracy

Smaller patches: Allow the network to see only little context

---

We want a good localization and the use of context at the same time



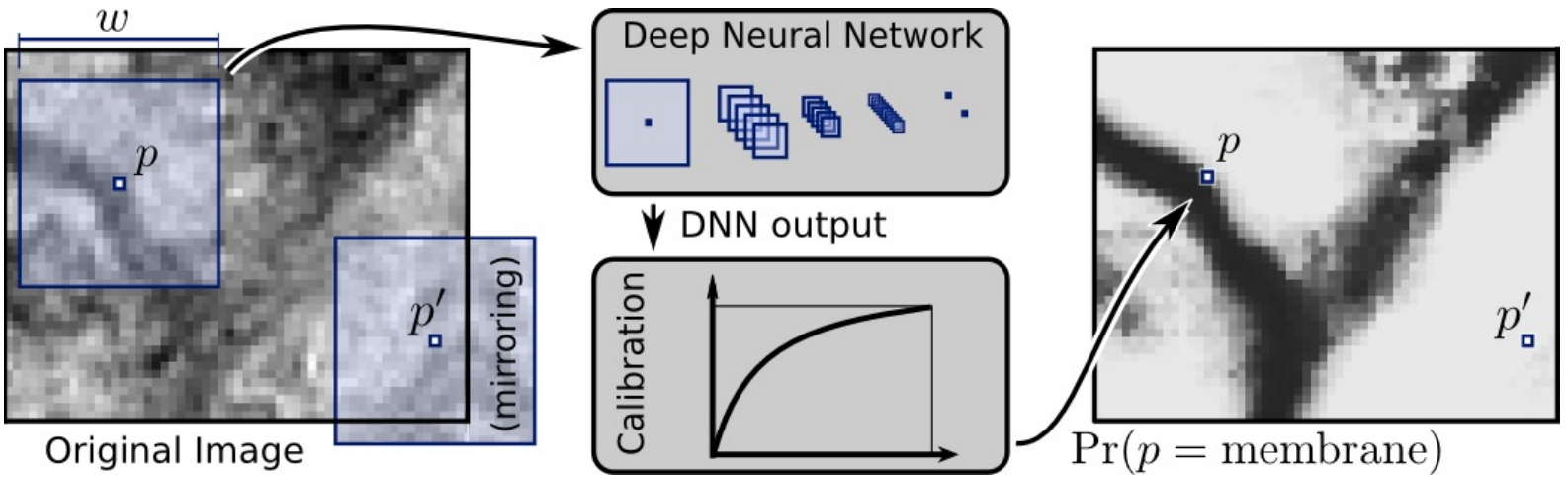
Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural net- works segment neuronal membranes in electron microscopy images. In: NIPS. pp. 2852{2860 (2012)

The winner (ISBI 2012)

- Calibration:

- The network outputs cannot be directly interpreted as probability values; instead, they tend to severely overestimate the membrane probability.
  - Compare all outputs obtained on T_{test} (a total of 2.6 million instances) to the ground truth, to compute the transformation relating the network output value and the actual probability of being a membrane;

- The resulting function is well approximated by a monotone cubic polynomial, whose coefficients are computed by least-squares fitting.

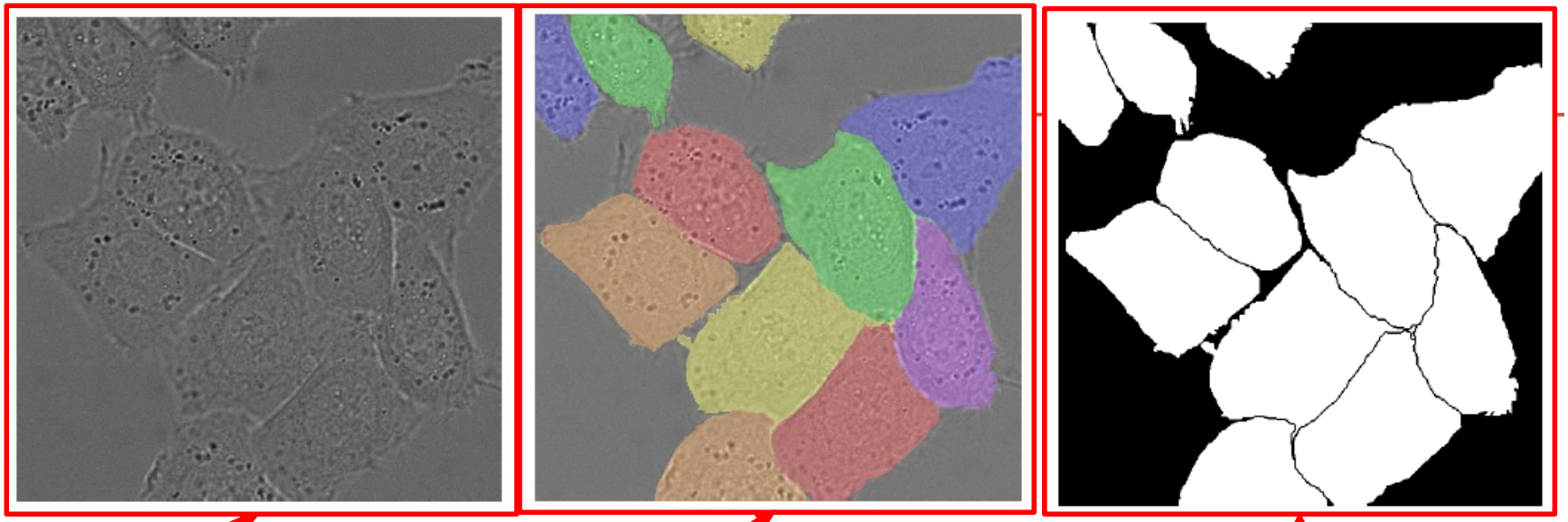


Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural net- works segment neuronal membranes in electron microscopy images. In: NIPS. pp. 2852{2860 (2012)

ISBI 2015 Task

ISBI 2015- separation of touching objects of the same class

- Light microscopic images (recorded by phase contrast microscopy)
- Part of the ISBI cell tracking challenge 2014 and 2015

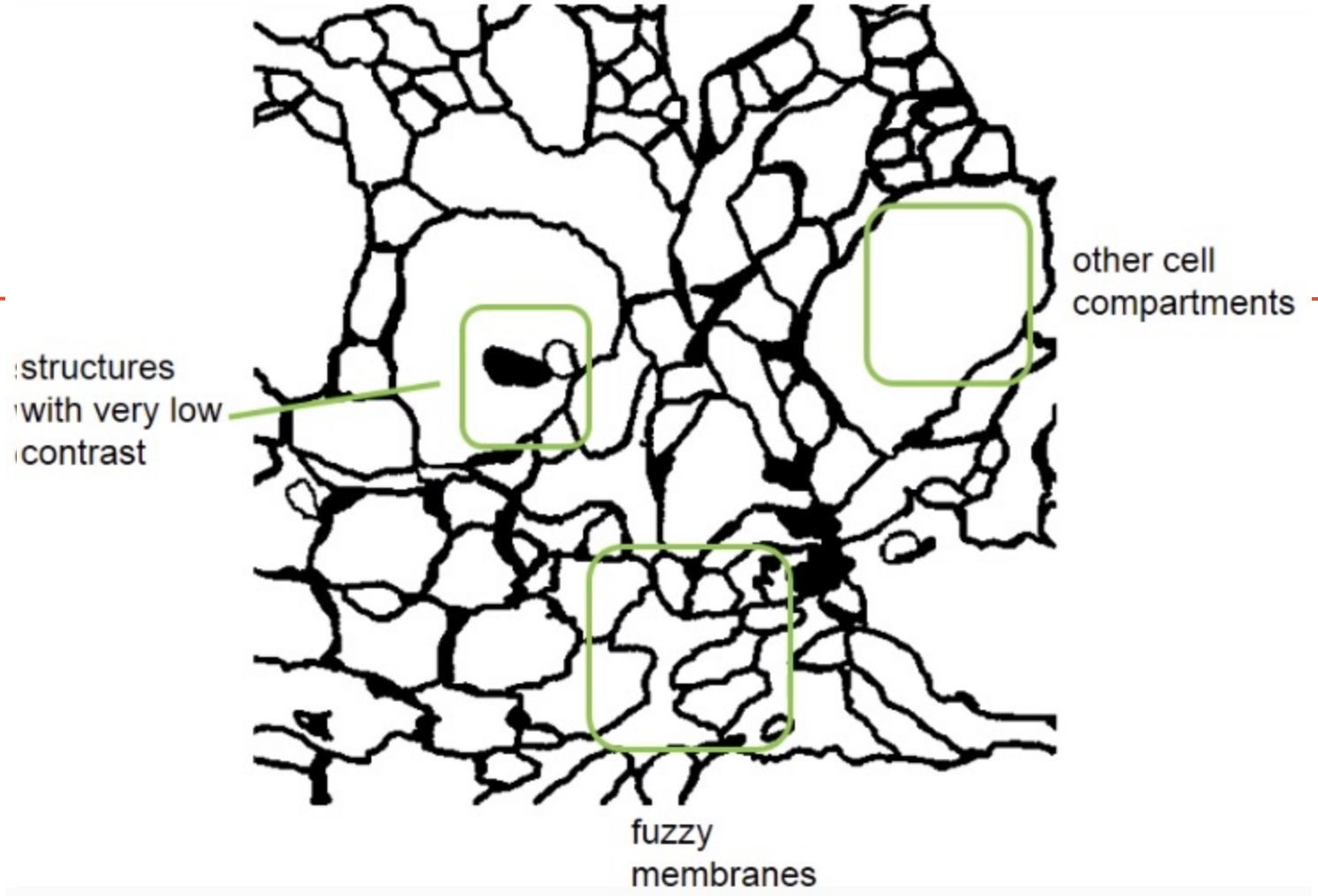


Raw image (HeLa cells)

Groundtruth segmentation.

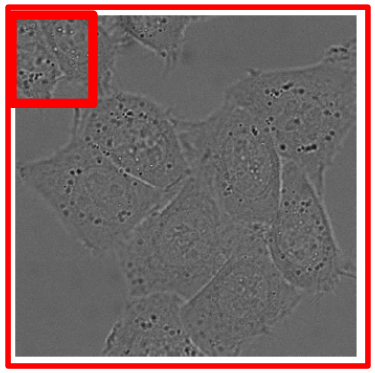
Generated segmentation mask (white: foreground, black: background)

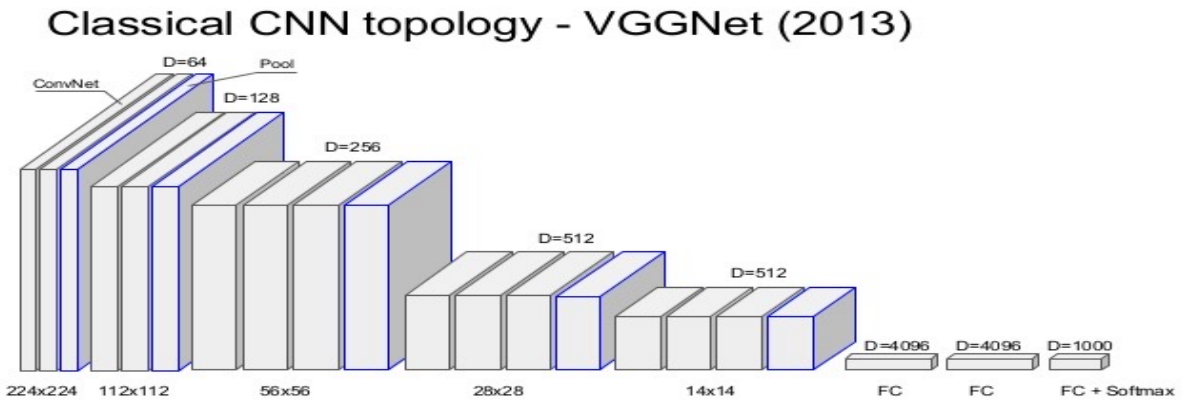
Challenge:
Segmentation of Neuronal Structures in EM



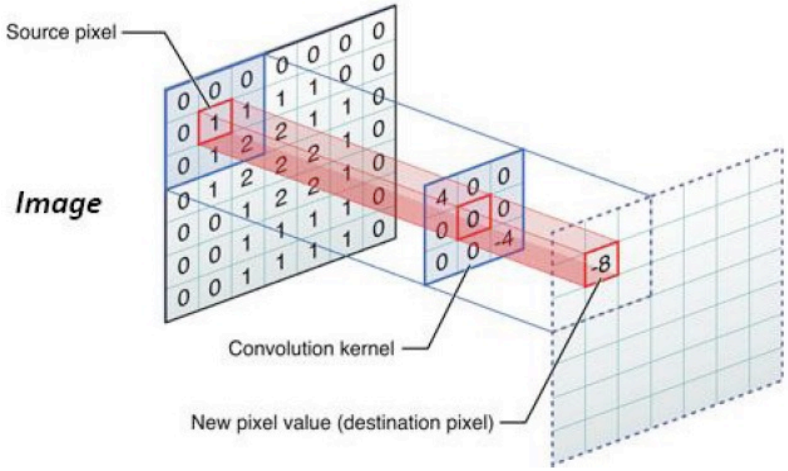
Challenge:
Segmentation of Neuronal Structures in EM

How to avoid the redundancy in the sliding window approach?





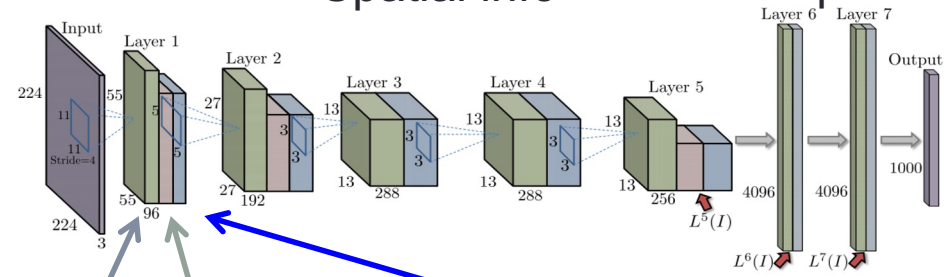
Remember: Convolutional Neural Network layers



Convolutional layer

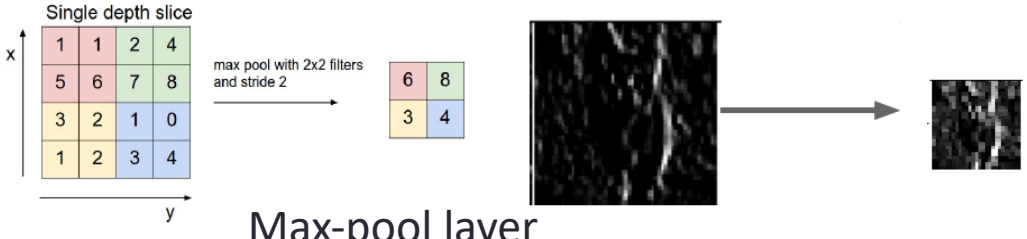
Spatial info

No spatial info



Filter Response

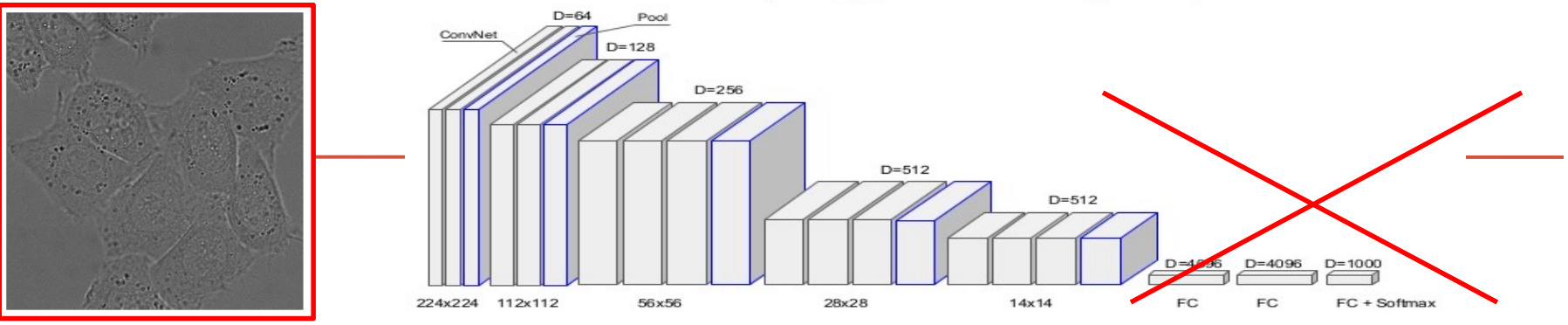
ReLU max(0,x)



Max-pool layer

Convolutional Neural Networks (CNN)

- The use of convolutional networks is on classification tasks where the output to typical image is a single class label.



- The desired output should include localization

A class label is supposed to be assigned to each pixel.

https://www.saagie.com/blog/object-detection-part1

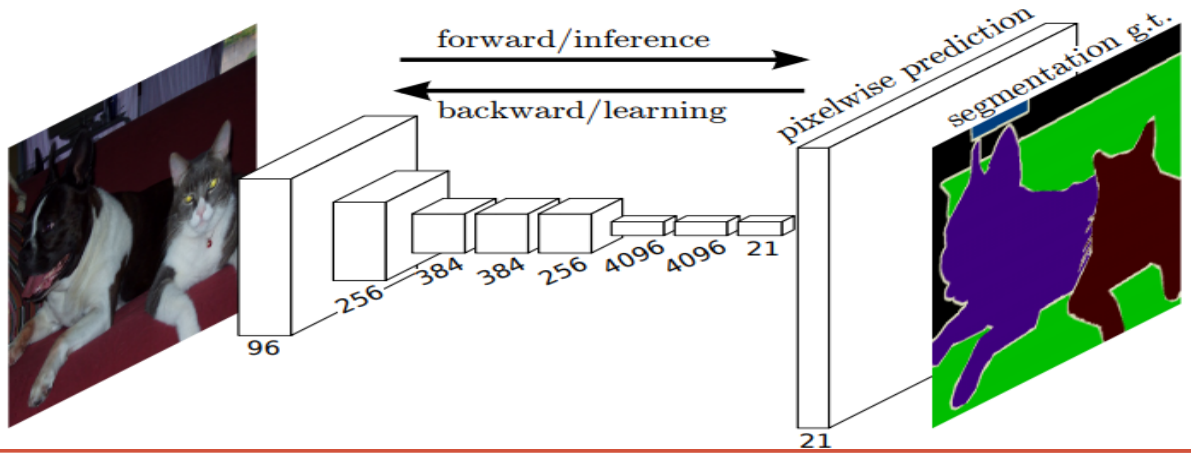
Biomedical Image Segmentation with U-net

- U-net learns segmentation in an end-to-end setting
- Touching objects of the same class
- Very few annotated images (approx. 30 per application)

IEEE International Symposium on Biomedical Imaging (ISBI 2015)

https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

Fully convolutional neural network



- **Localization and the use of context at the same time**
- **Input image with any size**
- **Added Simple Decoder (Upsampling + Conv)**
- **Removed all Dense Layers**

Localization and the use of context at the same time

https://www.azavea.com/blog/2017/05/30/deep-learning-on-aerial-imagery/

Recall: Convolutional Neural Network (CNN)

- A standard CNN for image classification is composed of:
  - Convolutional layers
  - Down-sampling layers
    - Stride convolution
    - Max pooling
    - Avg. Pooling
  - Batch normalization
  - Activation functions (e.g. ReLU)

Discrete convolution

- A discrete convolution is a linear transformation
  - Sparse – only few inputs contribute to a given output unit
  - Reuses parameters – same kernel is applied over multiple input elements

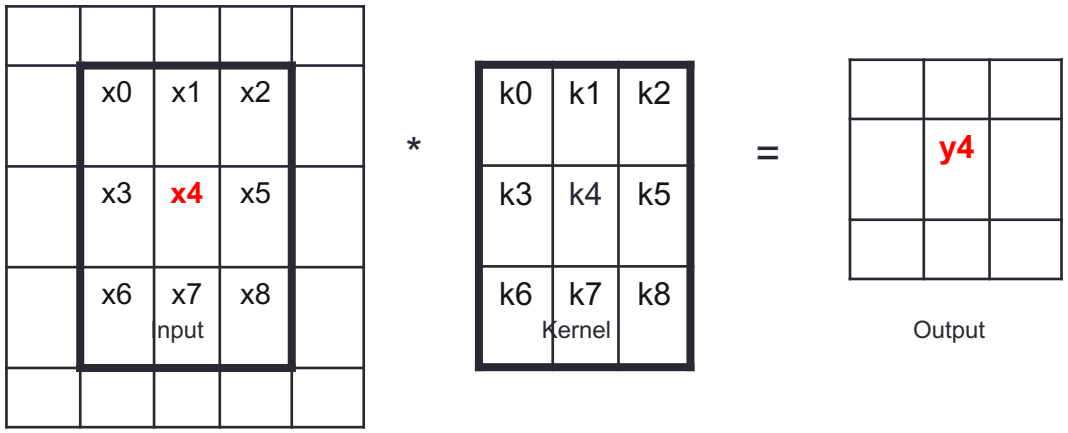


Figure: In this example, each output element is computed using 9 pixels

Convolution layer takes an input feature map of dimension $W \times H \times N$ and produces an output feature map of dimension $\widehat{W} \times \widehat{H} \times M$

Each layer is defined using following parameters:

Input channels (N)
Output channels (M)
Kernel size, Padding, Stride

Convolutional Layer

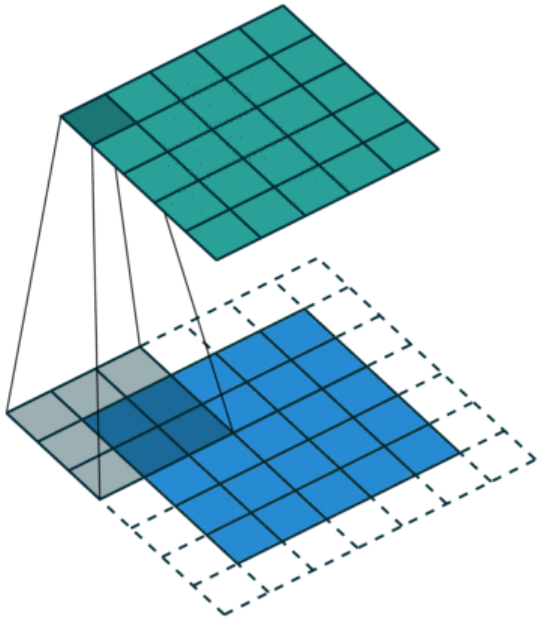


Figure: 5x5 input is convolved with 3x3 kernel with stride=padding=1 to produce an output of size 5x5.

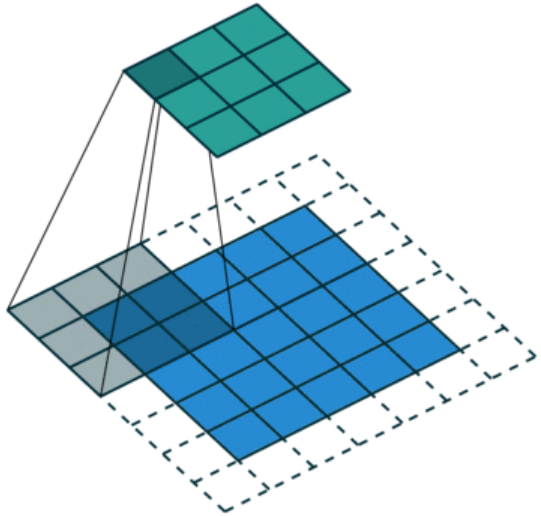
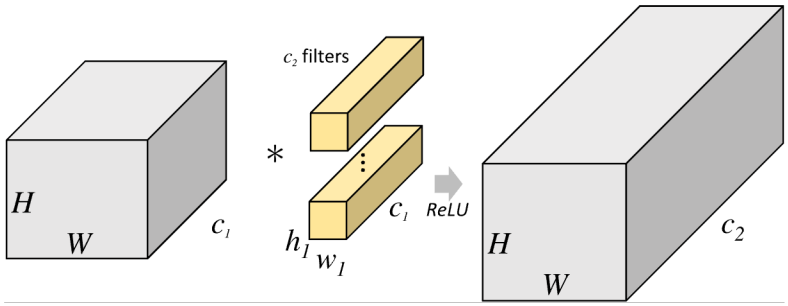


Figure: 5x5 input is convolved with 3x3 kernel with **stride=2** and padding=1 to produce an output of size 3x3.

Convolutional Layer

- Convolution layer takes C_1dimensional input feature map and produces a C_2 dimensional output feature

- Each layer is defined using the following parameters:
  - # Input channels (C_1)
  - # Output channels (C_2)
  - Kernel size ($w_1 \times h_1$)
  - Padding
  - Stride



- # of parameters learned by convolution layer is $\boldsymbol{n^2NM}$

Group Convolutional Layer

- Input and kernel are **split into g groups** across channel dimension
- Each group then performs the convolutions independently
- Each layer is defined using following parameters:
  - # Input channels (C_1)
  - # Output channels (C_2)
  - Kernel size ($w_1 \times h_1$)
  - Padding
  - Stride
  - Dilation rate (r)
  - # of groups (g)

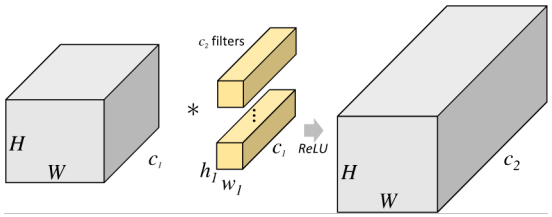


Figure: Standard convolution

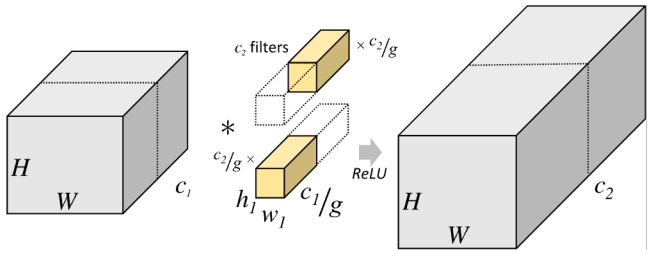
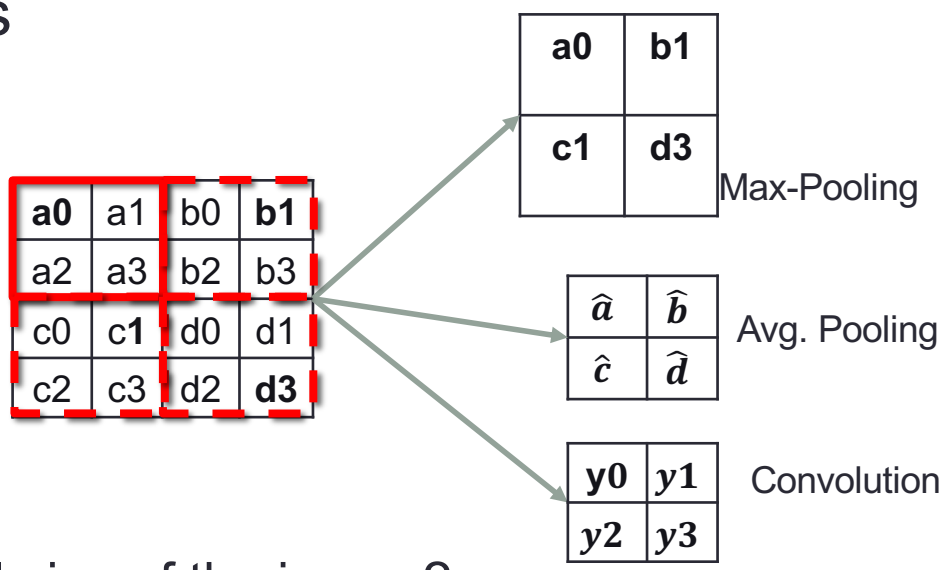


Figure: Grouped convolution

Down-sampling

- Learning representations at multiple scales is a fundamental step in computer vision
  - Laplacian Pyramids
  - SIFT, etc.
- Down-sampling in CNNs
  - Strided convolution
  - Max pooling
  - Avg. Pooling



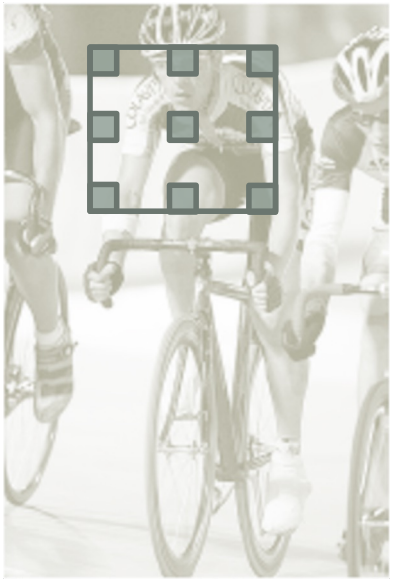
  - How to recover the original size of the image?

Solution: Dilation

- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?

Solution: Dilation

- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?



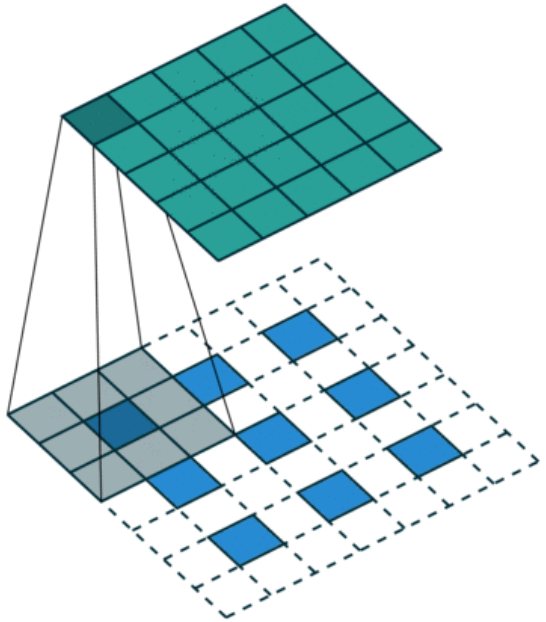
Solution: Dilation

- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?

Addressing the resolution problem

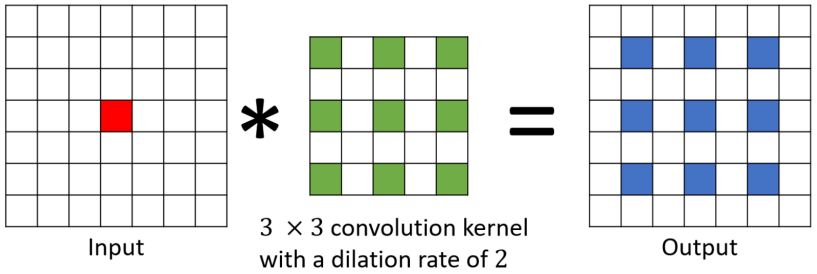
‘deconvolutional’ layers (backwards convolution).

- Additional memory and computational time.
- Learning additional parameters.

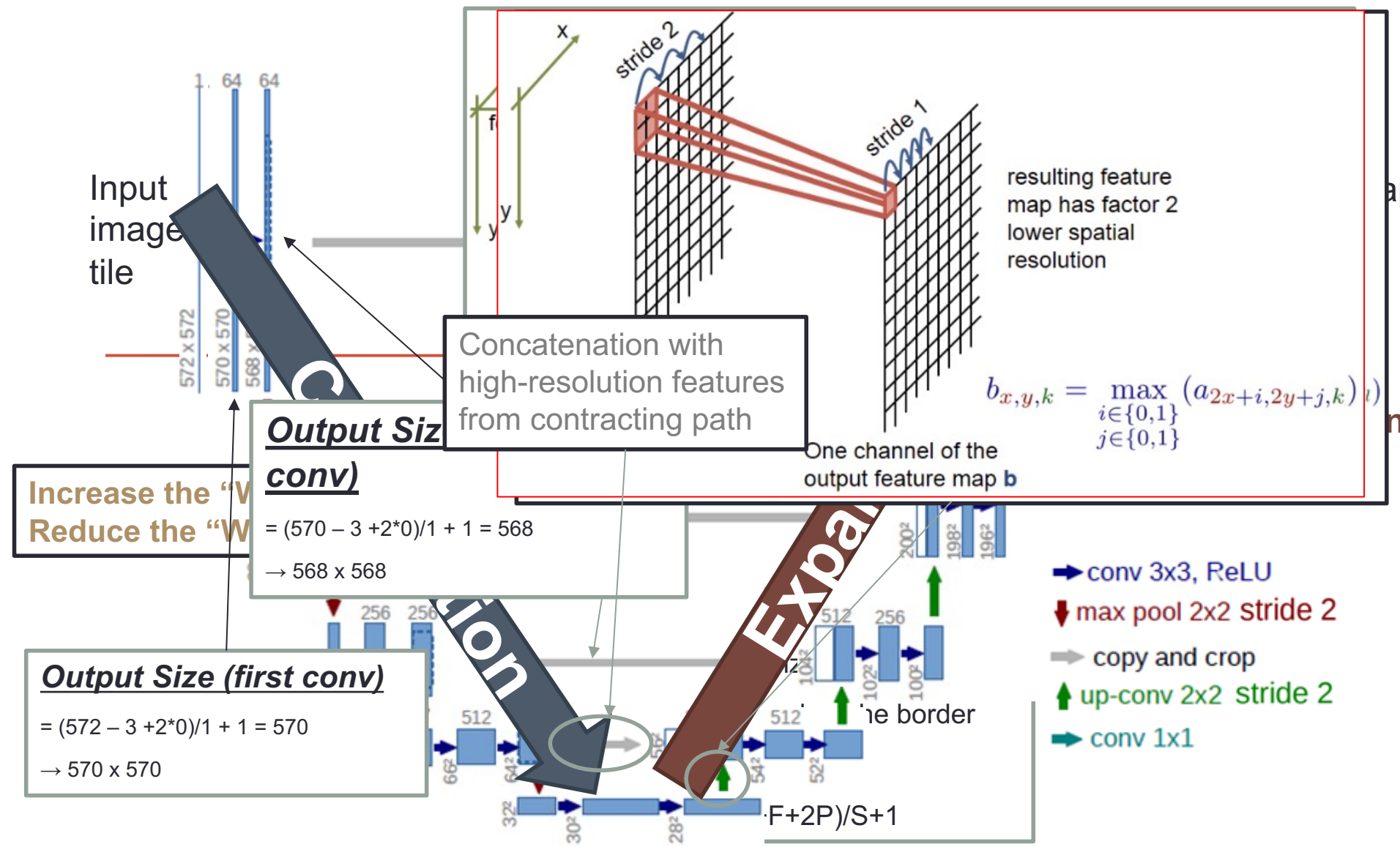


Dilated Convolution Layer

- Inserts spaces between the kernel element to increase the effective size of kernel
- Same as the convolutional layer except it has additional parameter, dilation rate, that controls the spacing
- Each layer is defined using following parameters:
  - # Input channels (C_1)
  - # Output channels (C_2)
  - Kernel size ($w_1 \times h_1$)
  - Padding
  - Stride
  - Dilation rate (r)



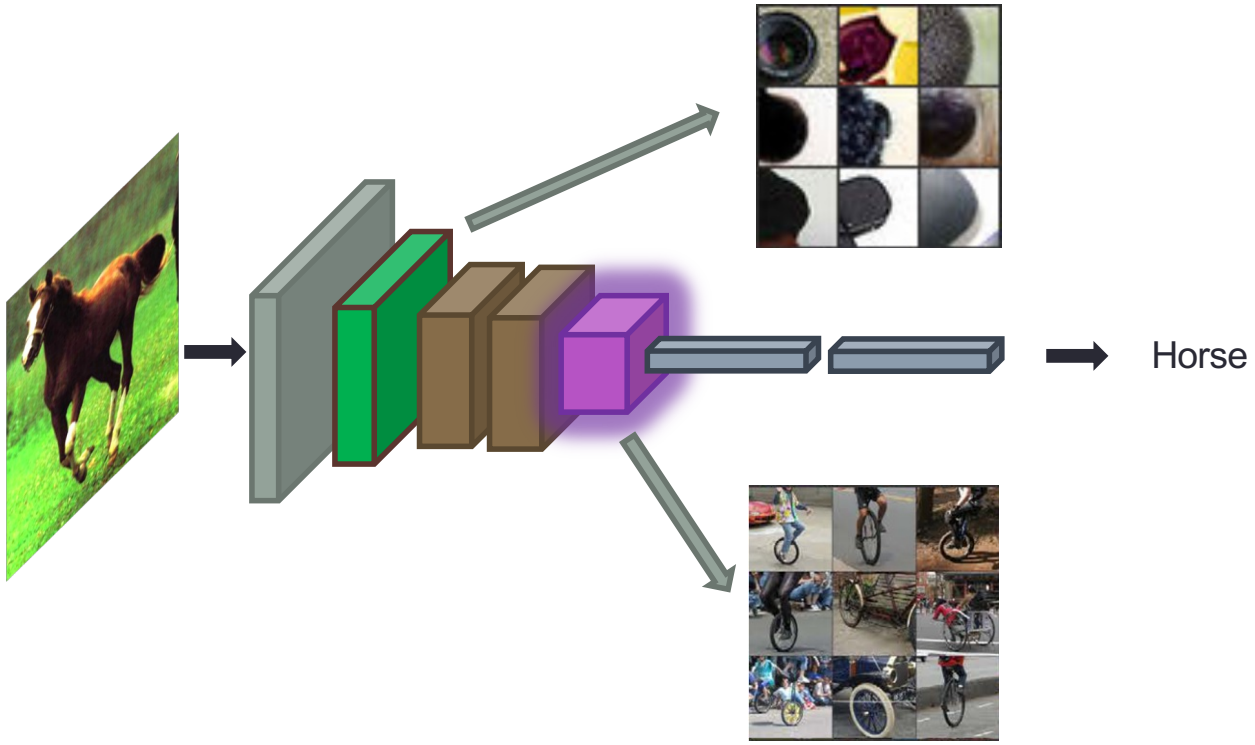
U-NET Architecture



http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

The resolution issue

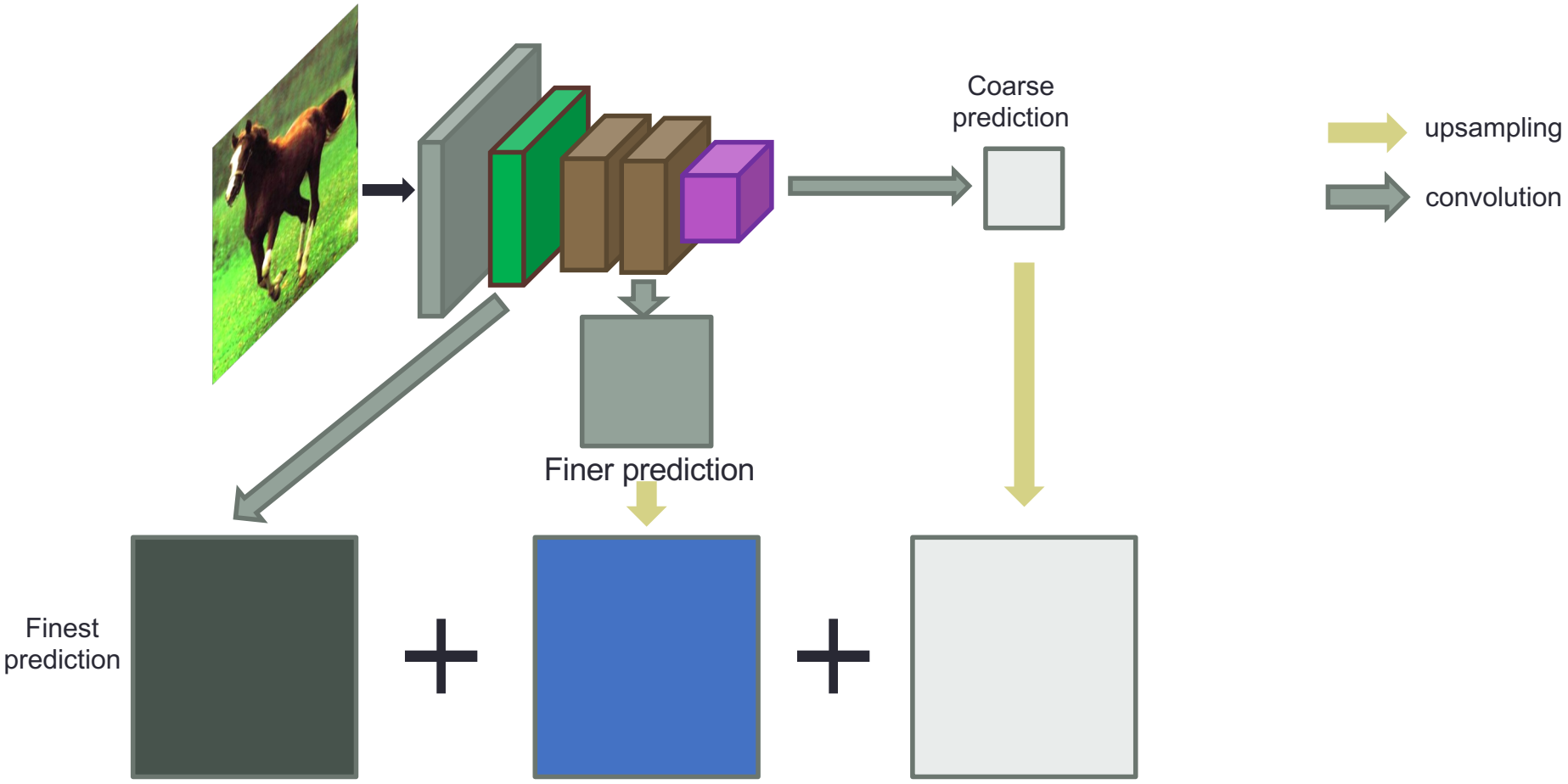
- Problem: early layers not semantic



- Late layers lose fine details information.

Visualizations from : M. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In *ECCV* 2014.

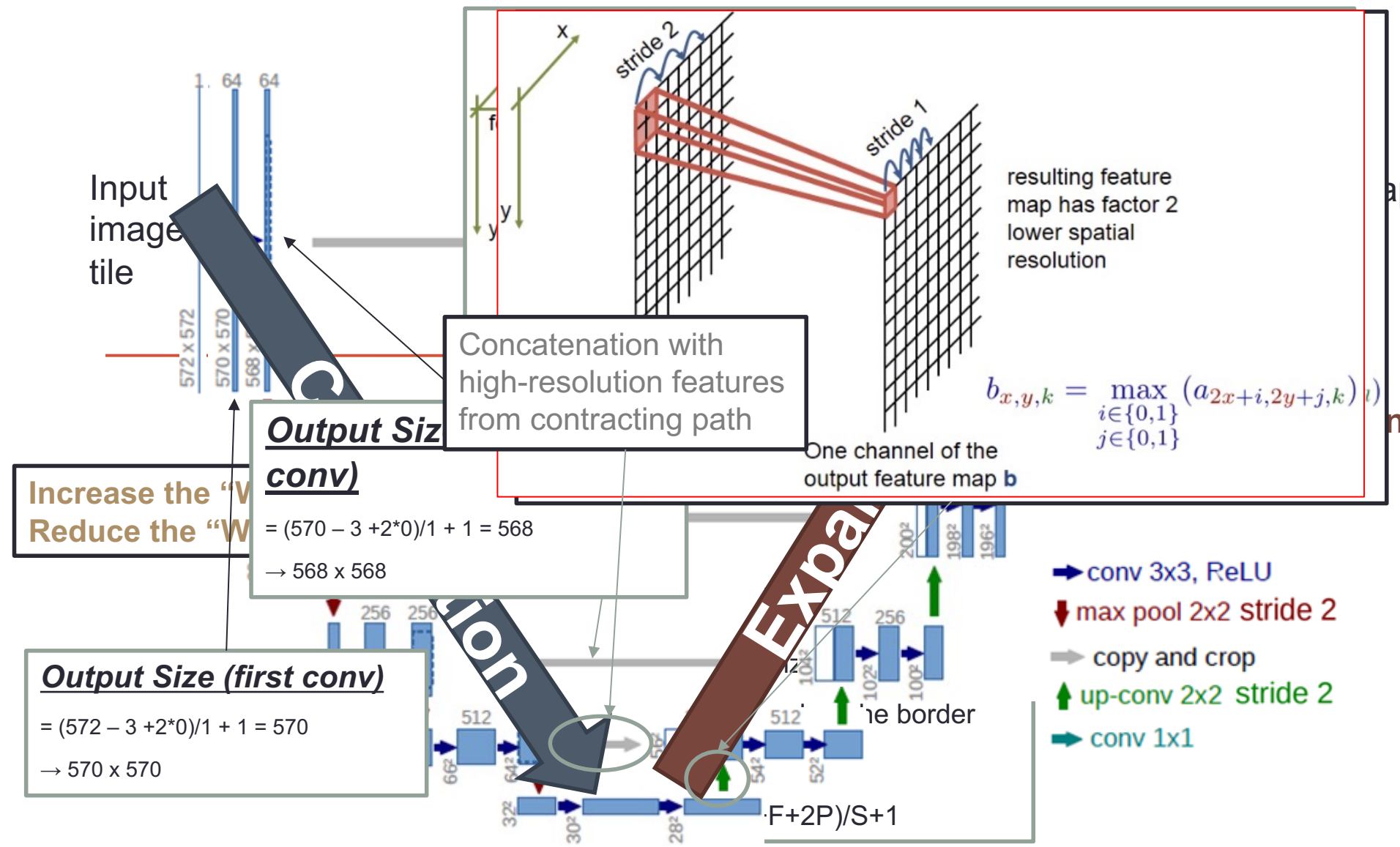
Solution: Skip connections



Skip connections

Fully convolutional networks for semantic segmentation. Evan Shelhamer, Jon Long, Trevor Darrell. In *CVPR* 2015

U-NET Architecture



http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

U-Net Training

Soft-max:

$$p_k(x) = \exp\big(a_k(x)\big) / \sum_{k'=1}^{K} \exp(a_{k'}(x))$$

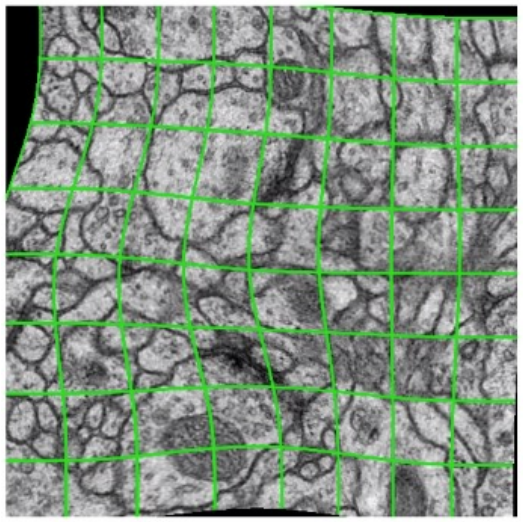
- k- Feature channel
- $a_k(x)$ - The activation in feature channel k at pixel position x

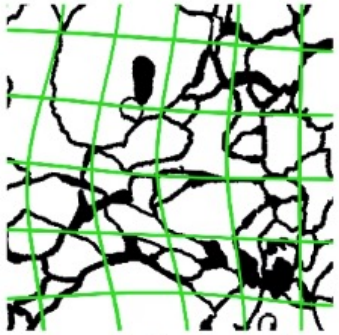
Cross-Entropy loss function:

$$E = -\sum_{x \in \Omega} w(x) log(\mathrm{p}_{\mathrm{l(x)}}(\mathrm{x}))$$

- $w(x)$- True label per a pixel

Data augmentation





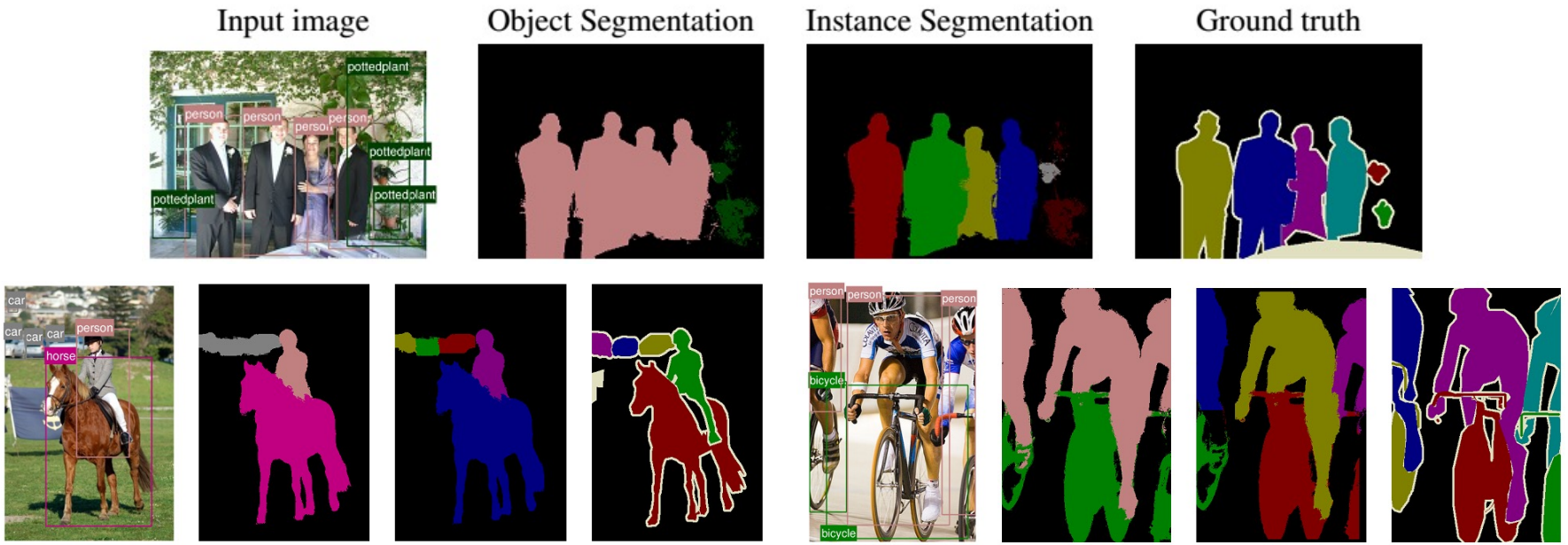
correspondingly deformed
manual labels

Augment Training Data using Deformation

- Random elastic deformation of the training samples.
- Shift and rotation invariance of the training samples.
- They use random displacement vectors on 3 by 3 grid.
- The displacement are sampled from Gaussian distribution with standard deviation of 10 pixels

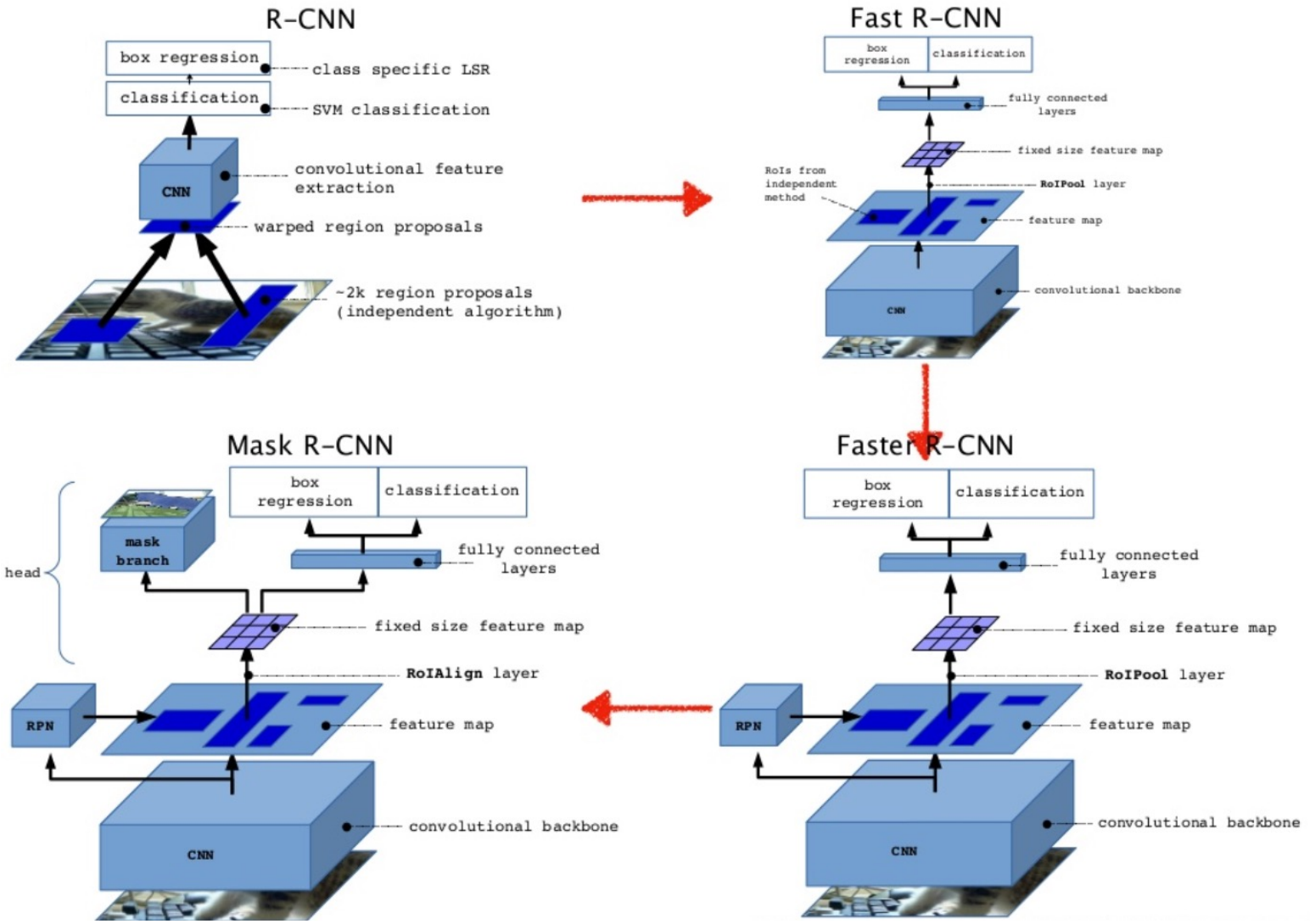
https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

Pedestrian segmentation



What about Instance and Panoptic Segmentation?

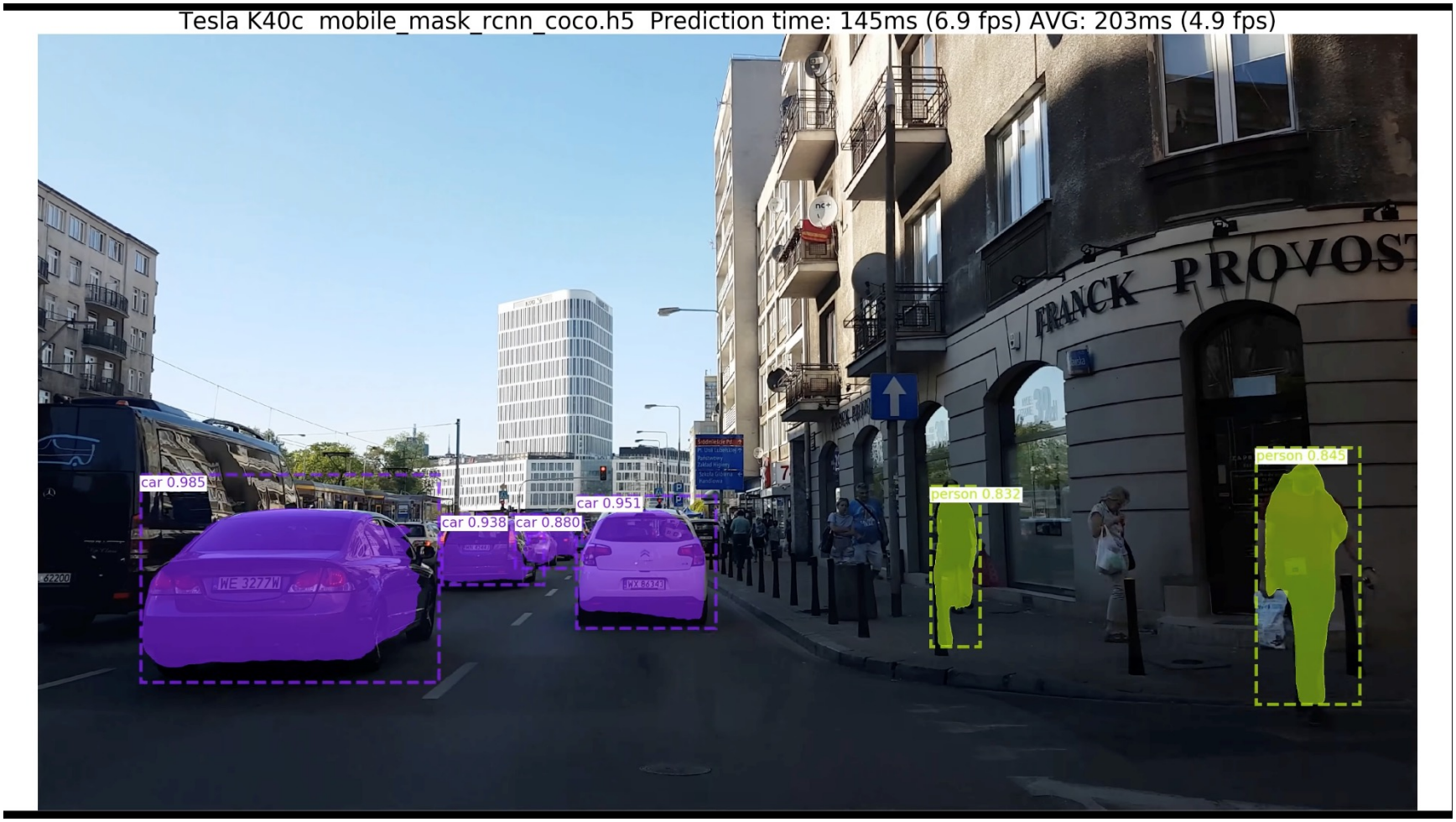
Instance segmentation



Mask-RCNN does: detection, classification and segmentation

City segmentation

Mobile Mask-RCNN



Original R-CNN

1. Input image
2. Extract region proposals (i.e with selective search)
3. Use transfer learning to compute features from each proposal.
4. Classify each proposal with Support Vector Machine

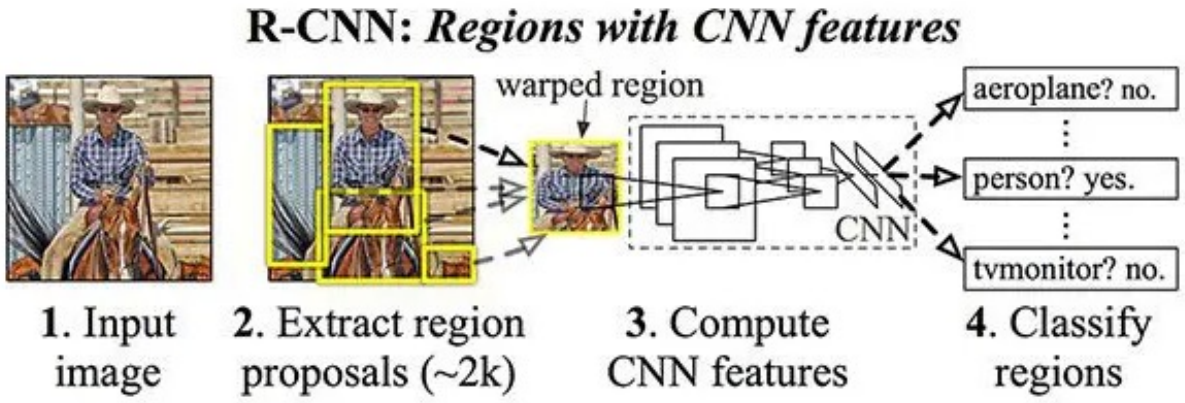


Figure 2: The original R-CNN architecture (source: Girshick et al,. 2013)

Fast R-CNN

1. Input image and ground truth bounding boxes
2. Extract feature maps
3. Apply ROI pooling to obtain ROI feature vector
4. Use two sets of FC layers

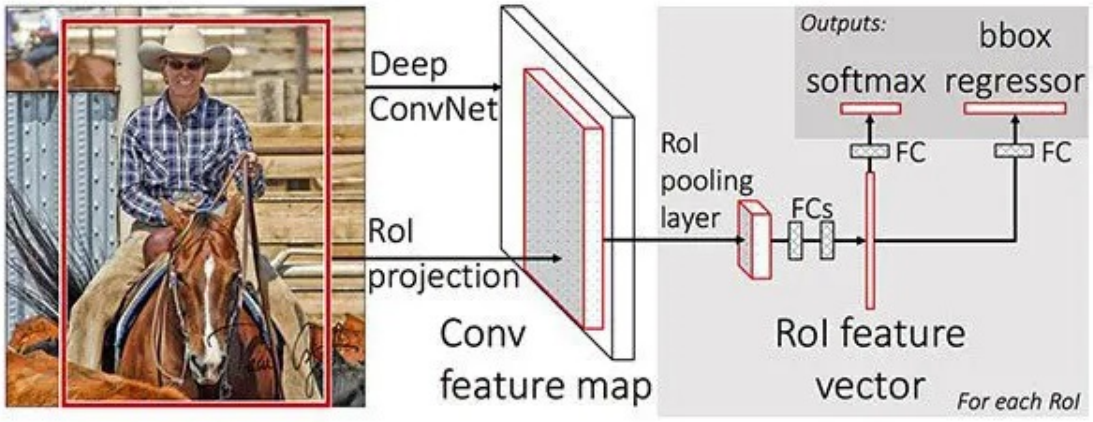
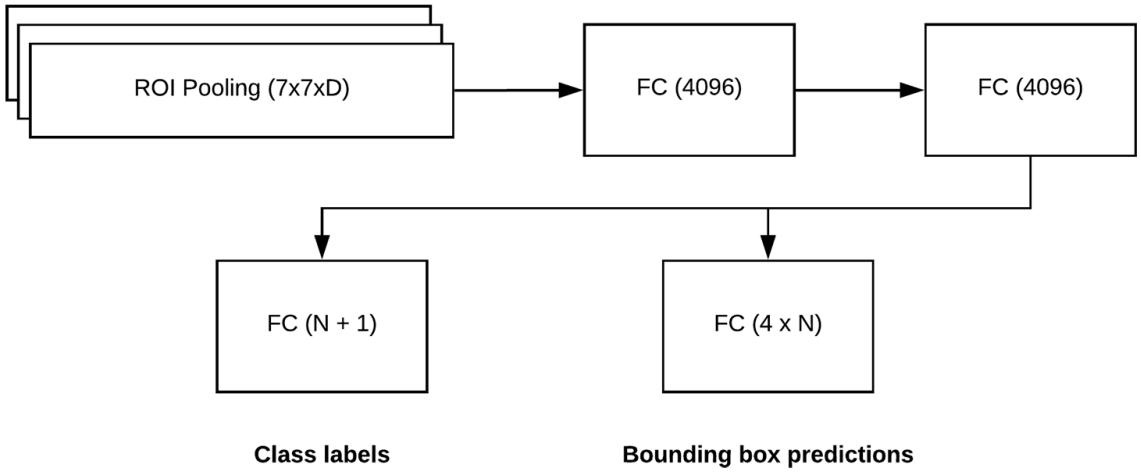


Figure 3: The Fast R-CNN architecture (source: Girshick et al., 2015).

The main benefit is that the network is end-to-end trainable

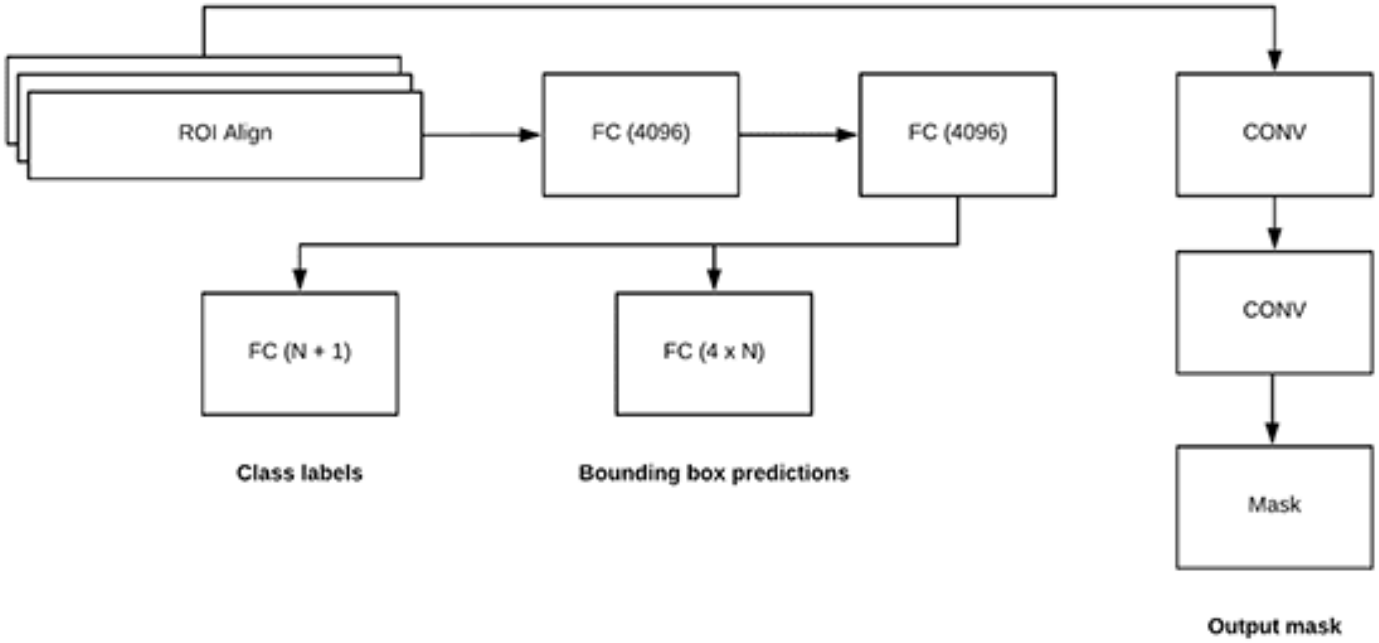
Faster R-CNN

- Takes the output of the ROI Pooling module and passes it through 2 FC layers + 2 final FC layers:
  - One FC layer is N+1-dimension: class labels + background
    → **class label probabilities**
  - One FC layer is 4 x N: deltas for the predicted bounding box
    → **bounding box predictions**



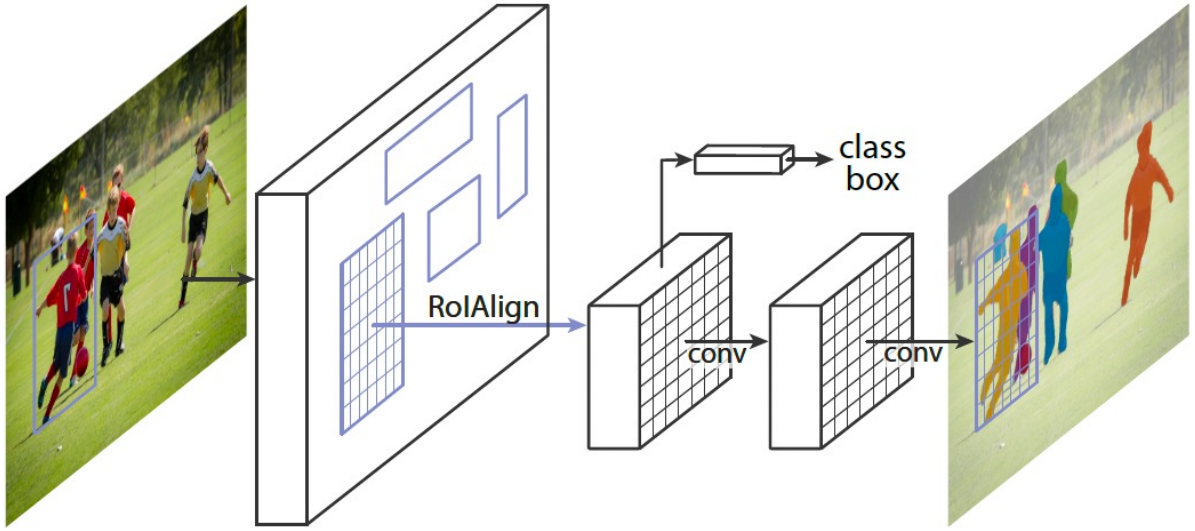
Mask R-CNN: Architecture

Replaces the ROI Pooling module with a more accurate ROI Align module.
Additional branch that predicts the actual mask of an object/class



Mask R-CNN: Architecture

Replaces the ROI Pooling module with a more accurate ROI Align module.
Additional branch that predicts the actual mask of an object/class



[He et al., 2017]

Figure 1. The **Mask R-CNN** framework for instance segmentation.

Mask R-CNN: Understanding the output

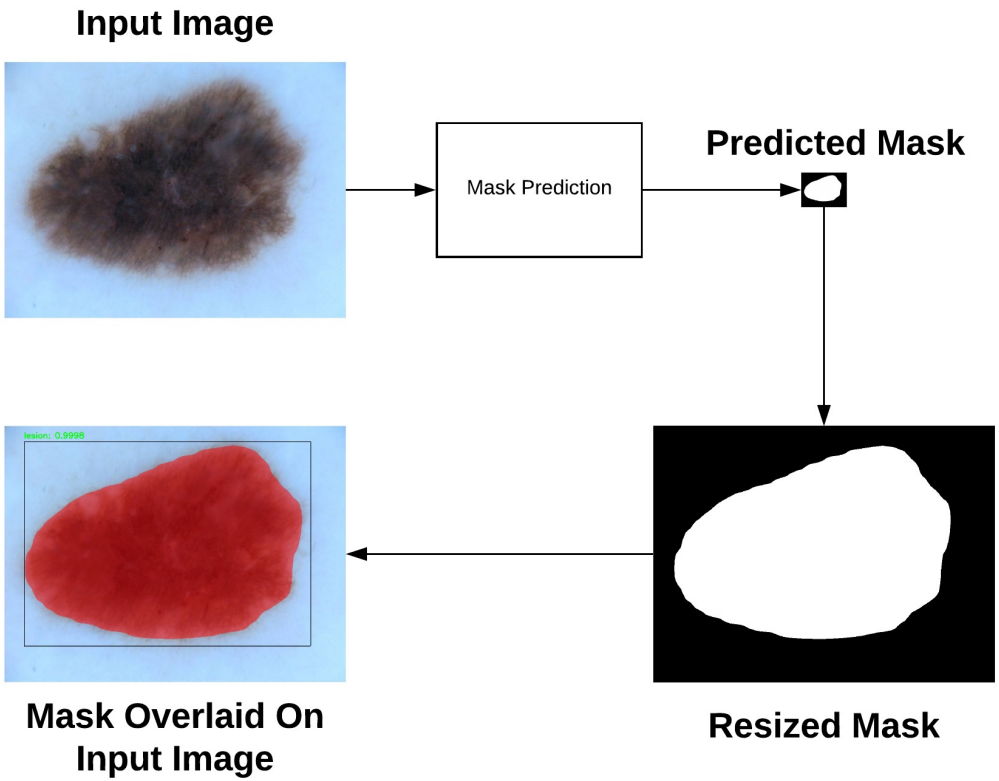
- Faster R-CNN/Mask R-CNN leverages a Region Proposal Network (RPN) to generate regions that potentially contain an object.
- Each of the regions is ranked based on its "*objectness score*" (i.e., how likely it is a given region could contain an object), and the top N regions are kept (N=300 based on He 2017).
- Each of the N most ROIs goes through the 3 branches
  - Label prediction
  - Bounding box prediction
  - Mask prediction

Mask R-CNN: Understanding the output

Example:

- Mask size is **15 x 15**
- During prediction, keep the top **100** detection boxes
- L is the number of class labels in the dataset (i.e, 90):
- Output from the mask module will be **100 x 90 x 15 x 15**
- After performing a forward pass we can loop over each of the detected bounding boxes, find the class label index with the largest corresponding probability, and use index lookup
- After performing the lookup we now have instanceMask, a 15 x15 mask that represents the pixel-wise segmentation of the i-th detected object.
- Scale the mask back to the original dimensions using nearest neighbor interpolation.
- Why nearest neighbor interpolation? nearest neighbor interpolation will preserve all original values of the image during resizing without introducing new values. While nearest neighbor interpolation is less appealing to the human eye, such as bilinear or bicubic interpolation, for example, the benefit here is that we do not introduce “new” object/mask assignments when resizing the image.

Mask R-CNN: Understanding the output



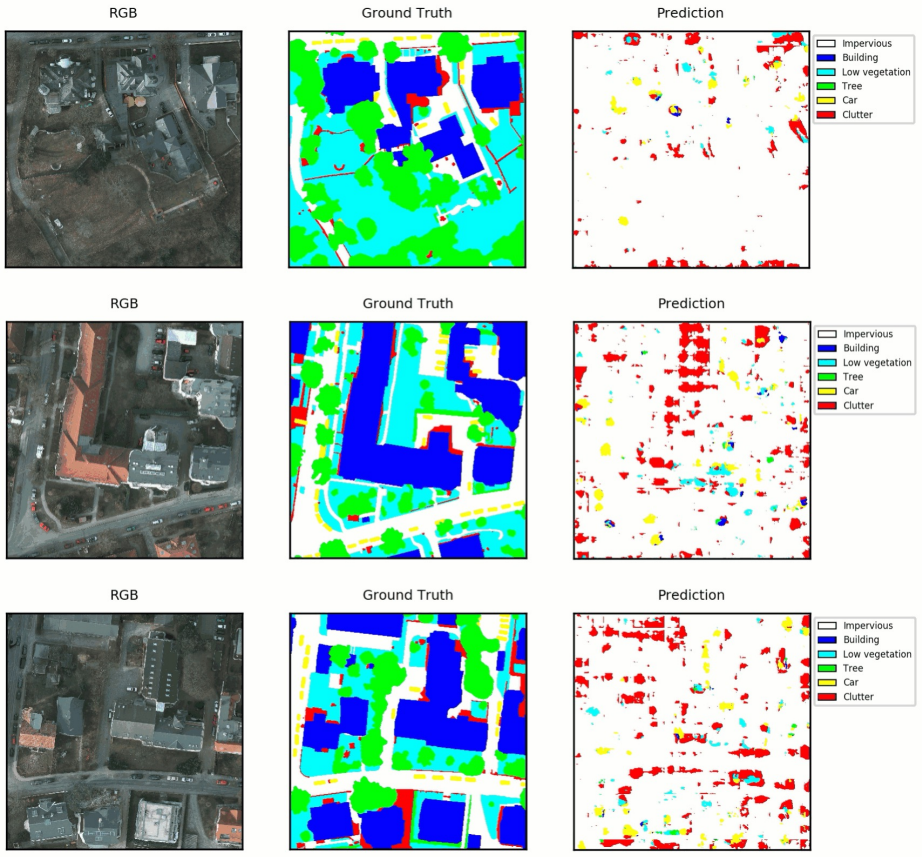
An example of extracting a 15⇥15 mask, resizing to the original image dimensions, and then overlaying the mask on the detected object.

We start with our input image and then feed it through our Mask R-CNN network to obtain our mask prediction. The predicted mask is only 15 x 15 pixels so we apply nearest neighbor interpolation to resize it back to the original image dimensions. The resized mask can then be overlaid on the original input image.

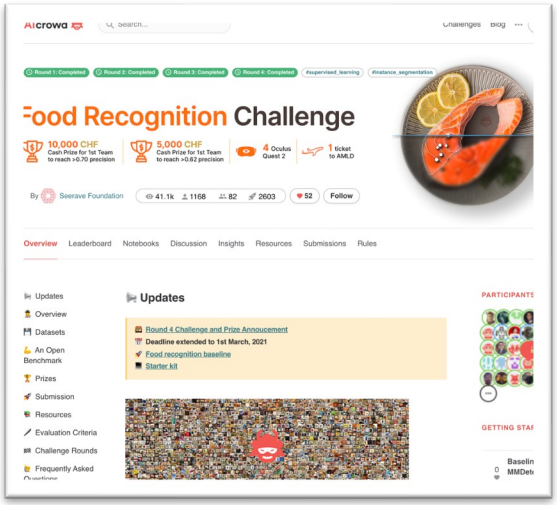
Other applications – Satellite image segmentation (Raster Vision)

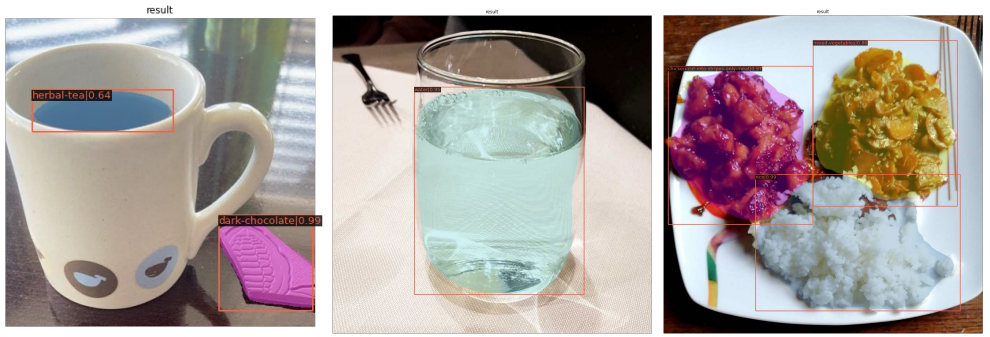
- Aerial and satellite imagery gives us the unique ability to look down and see the earth from above.
  - Used to measure deforestation,
  - map damaged areas after natural disasters,
  - spot looted archaeological sites, and many more current and untapped use cases.
- Enormous and ever-growing amount of imagery presents a significant challenge: how can we derive value and insights from all of this data?

  - There are not enough people to look at all of the images all of the time.

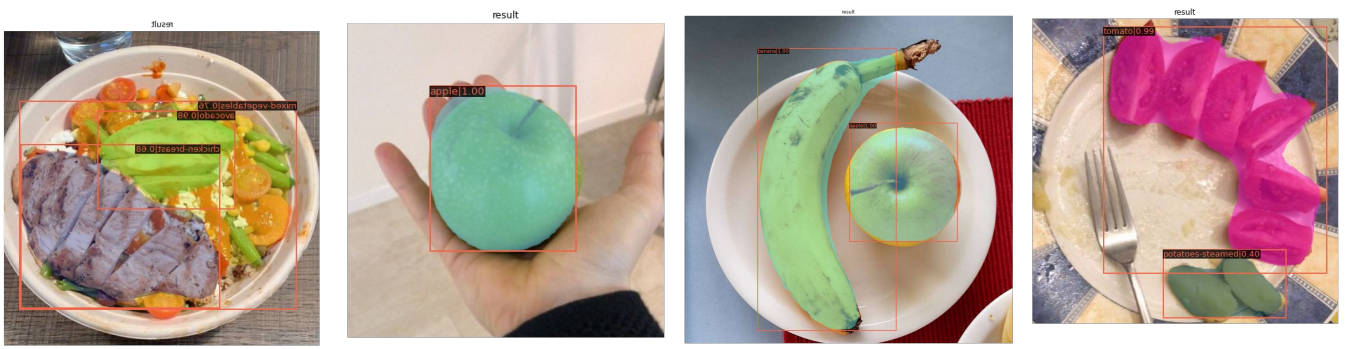


Food segmentation





AI crowd dataset 26000 annotated segmented images with 273 classes



Summary and Conclusion

U-net advantages

- Flexible and can be used for any rational image masking task
- High accuracy (given proper training, dataset, and training time)
- Doesn’t contain any fully connected layers
- Faster than the sliding-window (1-sec per image)
- Proven to be very powerful segmentation tool in scenarios with limited data
- Succeeds to achieve very good performances on different biomedical segmentation applications.

U-net disadvantages

- Larger images need high GPU memory.
- Takes significant amount of time to train (relatively many layers)
- Pre-trained models not widely available (it's too task specific)

Mask-RCNN – Still SoA for Instance segmentation