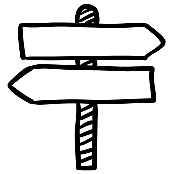
CSC 578
Neural Networks and Deep Learning

6-3. Convolutional Neural Networks (3)

Learning in CNN



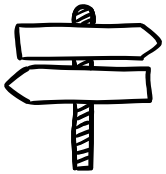
Cross-correlation

$$(I \otimes K)_{ij} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} I(i+m, j+n)K(m,n) \tag{1}$$

Convolution

$$\begin{aligned}(I * K)_{ij} &= \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} I(i-m, j-n)K(m,n) &(2)\\ &= \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} I(i+m, j+n)K(-m,-n) &(3)\end{aligned}$$

Learning in CNN



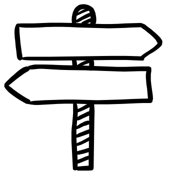
Convolution Output

$$(I * K)_{ij} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} \sum_{c=1}^{C} K_{m,n,c} \cdot I_{i+m,j+n,c} + b \tag{4}$$

Convolution Output Simplified

$$(I * K)_{ij} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} K_{m,n} \cdot I_{i+m,j+n} + b \tag{5}$$

Learning in CNN formulation



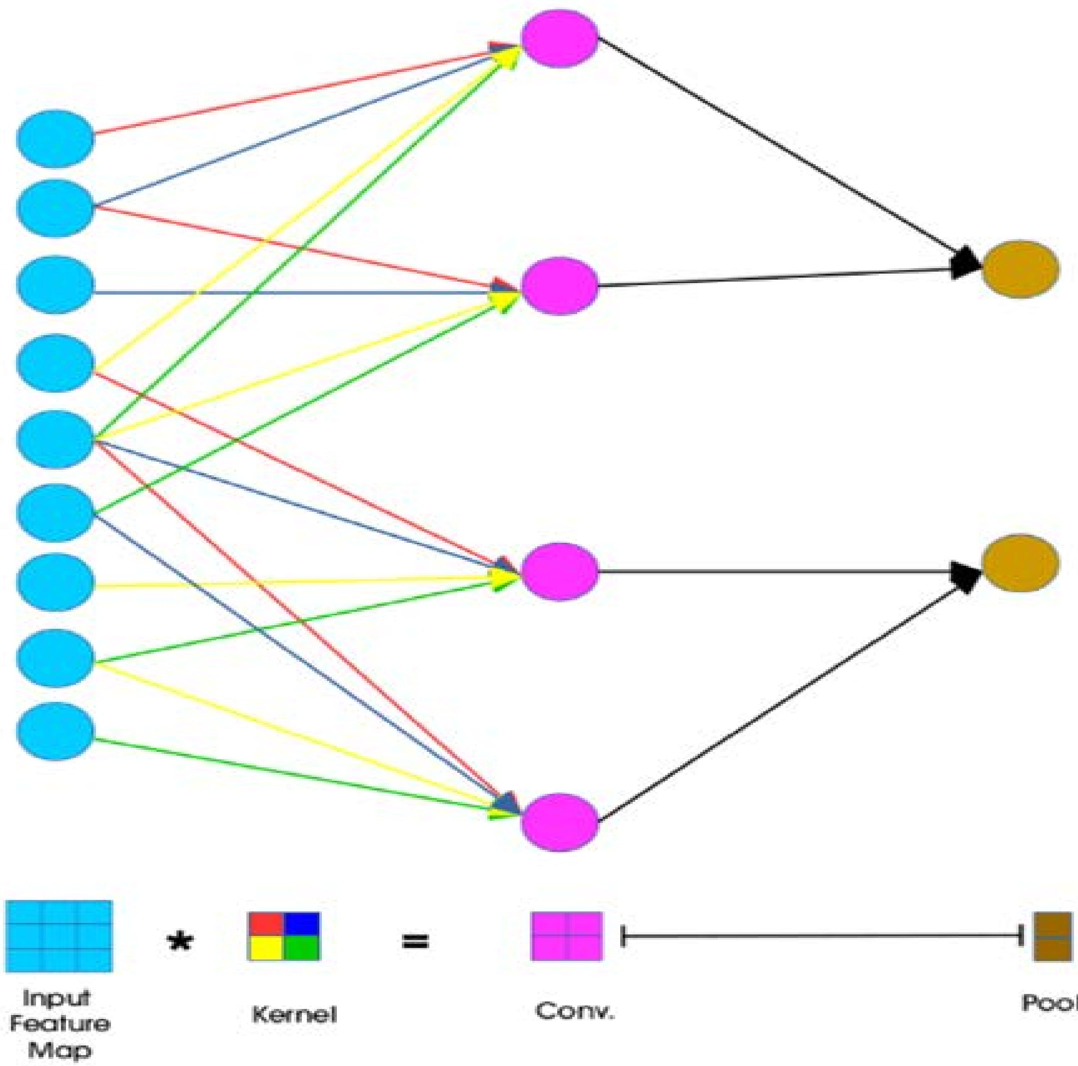
1. l is the l^{th} layer where $l = 1$ is the first layer and $l = L$ is the last layer.
2. Input x is of dimension $H \times W$ and has i by j as the iterators
3. Filter or kernel w is of dimension $k_1 \times k_2$ has m by n as the iterators
4. $w^l_{m,n}$ is the weight matrix connecting neurons of layer l with neurons of layer $l - 1$.
5. b^l is the bias unit at layer l.
6. $x^l_{i,j}$ is the convolved input vector at layer l plus the bias represented as

$$x^l_{i,j} = \sum_m \sum_n w^l_{m,n} o^{l-1}_{i+m,j+n} + b^l$$

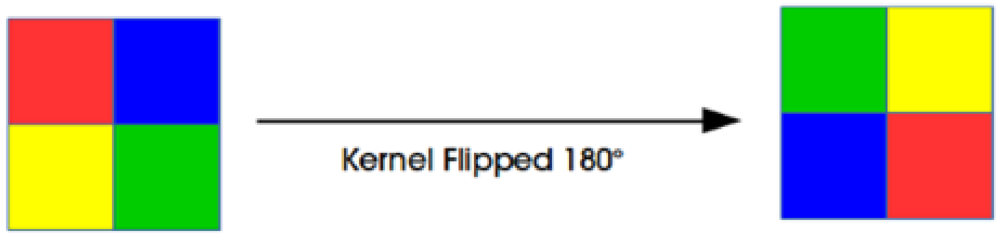
7. $o^l_{i,j}$ is the output vector at layer l given by

$$o^l_{i,j} = f(x^l_{i,j})$$

8. $f(\cdot)$ is the activation function. Application of the activation layer to the convolved input vector at layer l is given by $f(x^l_{i,j})$


*
=
Input
Feature
Map
Kernel
Conv.
Pool

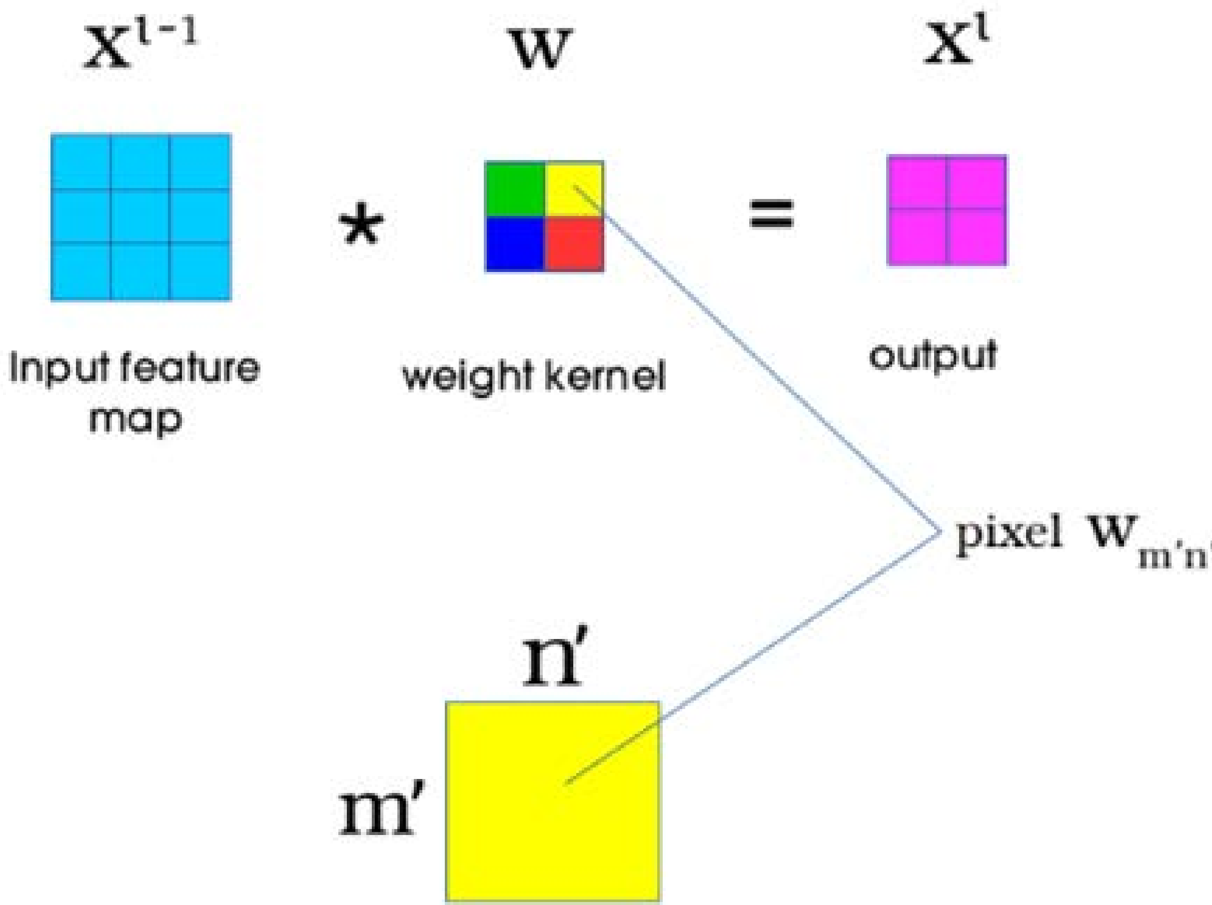
Forward Propagation



$$x_{i,j}^{l} = \mathrm{rot}_{180^\circ}\left\{w_{m,n}^{l}\right\} * o_{i,j}^{l-1} + b_{i,j}^{l} \tag{6}$$

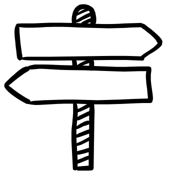
$$x_{i,j}^{l} = \sum_{m}\sum_{n} w_{m,n}^{l} o_{i+m,j+n}^{l-1} + b_{i,j}^{l} \tag{7}$$

$$o_{i,j}^{l} = f(x_{i,j}^{l}) \tag{8}$$



X^{l-1}
w
X^{l}
*
=
Input feature map
weight kernel
output
pixel $w_{m'n'}$
n'
m'

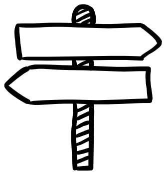
CNN BackPropagation



$$\frac{\partial E}{\partial w^l_{m',n'}} = \sum_{i=0}^{H-k_1} \sum_{j=0}^{W-k_2} \frac{\partial E}{\partial x^l_{i,j}} \frac{\partial x^l_{i,j}}{\partial w^l_{m',n'}} = \sum_{i=0}^{H-k_1} \sum_{j=0}^{W-k_2} \delta^l_{i,j} \frac{\partial x^l_{i,j}}{\partial w^l_{m',n'}} \tag{10}$$

$$\frac{\partial x^l_{i,j}}{\partial w^l_{m',n'}} = \frac{\partial}{\partial w^l_{m',n'}} \left(\sum_m \sum_n w^l_{m,n} o^{l-1}_{i+m,j+n} + b^l \right) \tag{11}$$

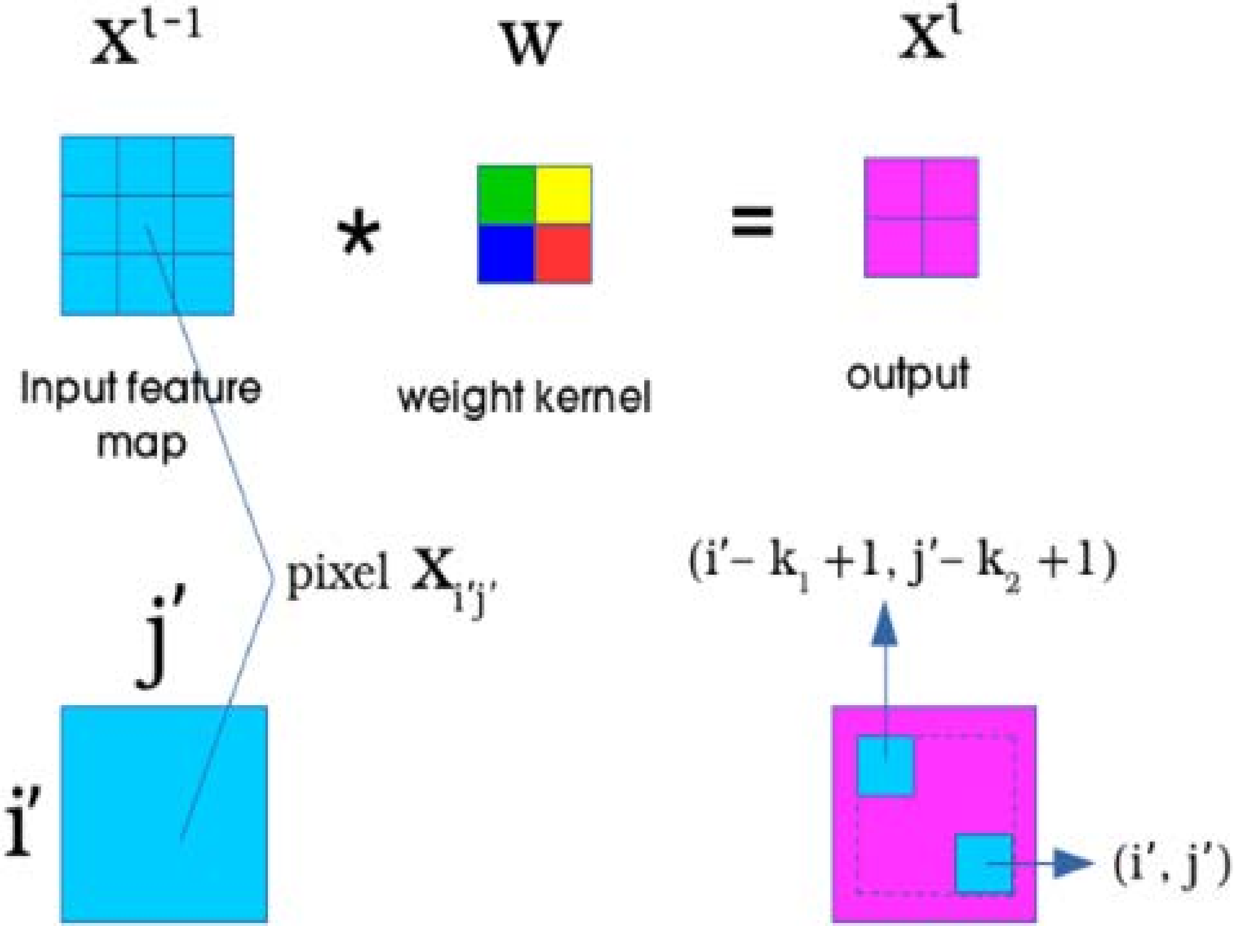
CNN BackPropagation



$$\begin{aligned}\frac{\partial x^l_{i,j}}{\partial w^l_{m',n'}} &= \frac{\partial}{\partial w^l_{m',n'}}\left(w^l_{0,0}\, o^{l-1}_{i+0,j+0} + \cdots + w^l_{m',n'}\, o^{l-1}_{i+m',j+n'} + \cdots + b^l\right)\\ &= \frac{\partial}{\partial w^l_{m',n'}}\left(w^l_{m',n'}\, o^{l-1}_{i+m',j+n'}\right)\\ &= o^{l-1}_{i+m',j+n'} \end{aligned} \tag{12}$$

$$\frac{\partial E}{\partial w^l_{m',n'}} = \sum_{i=0}^{H-k_1}\sum_{j=0}^{W-k_2} \delta^l_{i,j}\, o^{l-1}_{i+m',j+n'} \tag{13}$$

$$= \mathrm{rot}_{180^\circ}\left\{\delta^l_{i,j}\right\} * o^{l-1}_{m',n'} \tag{14}$$



X^{l-1}
W
X^{l}
*
=
Input feature map
weight kernel
output
pixel X_{i'j'}
j'
i'
(i'− k_1 +1, j'− k_2 +1)
(i', j')

From the diagram above, we can see that region in the output affected by pixel $x_{i',j'}$ from the input is the region in the output bounded by the dashed lines where the top left corner pixel is given by $(i'-k_1+1, j'-k_2+1)$ and the bottom right corner pixel is given by (i',j').

Using chain rule and introducing sums give us the following equation:

$$\begin{aligned}\frac{\partial E}{\partial x^l_{i',j'}} &= \sum_{i,j \in Q} \frac{\partial E}{\partial x^{l+1}_Q} \frac{\partial x^{l+1}_Q}{\partial x^l_{i',j'}} \\ &= \sum_{i,j \in Q} \delta^{l+1}_Q \frac{\partial x^{l+1}_Q}{\partial x^l_{i',j'}}\end{aligned} \tag{16}$$

Q in the summation above represents the output region bounded by dashed lines and is composed of pixels in the output that are affected by the single pixel $x_{i',j'}$ in the input feature map. A more formal way of representing Eq. 16 is:

$$\begin{aligned}\frac{\partial E}{\partial x^l_{i',j'}} &= \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} \frac{\partial E}{\partial x^{l+1}_{i'-m,j'-n}} \frac{\partial x^{l+1}_{i'-m,j'-n}}{\partial x^l_{i',j'}} \\ &= \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} \delta^{l+1}_{i'-m,j'-n} \frac{\partial x^{l+1}_{i'-m,j'-n}}{\partial x^l_{i',j'}}\end{aligned} \tag{17}$$

In the region Q, the height ranges from $i' - 0$ to $i' - (k_1 - 1)$ and width $j' - 0$ to $j' - (k_2 - 1)$. These two can simply be represented by $i' - m$ and $j' - n$ in the summation since the iterators m and n exists in the following similar ranges from $0 \leq m \leq k_1 - 1$ and $0 \leq n \leq k_2 - 1$.

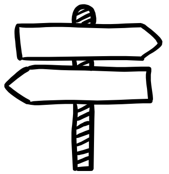
In Eq. 17, $x^{l+1}_{i'-m,j'-n}$ is equivalent to $\sum_{m'} \sum_{n'} w^{l+1}_{m',n'} o^{l}_{i'-m+m',j'-n+n'} + b^{l+1}$ and expanding this part of the equation gives us:

$$
\begin{aligned}
\frac{\partial x^{l+1}_{i'-m,j'-n}}{\partial x^{l}_{i',j'}} &= \frac{\partial}{\partial x^{l}_{i',j'}} \left(\sum_{m'} \sum_{n'} w^{l+1}_{m',n'} o^{l}_{i'-m+m',j'-n+n'} + b^{l+1} \right) \\
&= \frac{\partial}{\partial x^{l}_{i',j'}} \left(\sum_{m'} \sum_{n'} w^{l+1}_{m',n'} f\left(x^{l}_{i'-m+m',j'-n+n'} \right) + b^{l+1} \right)
\end{aligned} \tag{18}
$$

Further expanding the summation in Eq. 17 and taking the partial derivatives for all the components results in zero values for all except the components where $m' = m$ and $n' = n$ so that $f\left(x^l_{i'-m+m',j'-n+n'}\right)$ becomes $f\left(x^l_{i',j'}\right)$ and $w^{l+1}_{m',n'}$ becomes $w^{l+1}_{m,n}$ in the relevant part of the expanded summation as follows:

$$\begin{aligned}
\frac{\partial x^{l+1}_{i'-m,j'-n}}{\partial x^l_{i',j'}} &= \frac{\partial}{\partial x^l_{i',j'}}\left(w^{l+1}_{m',n'} f\left(x^l_{0-m+m',0-n+n'}\right) + \cdots + w^{l+1}_{m,n} f\left(x^l_{i',j'}\right) + \cdots + b^{l+1}\right) \\
&= \frac{\partial}{\partial x^l_{i',j'}}\left(w^{l+1}_{m,n} f\left(x^l_{i',j'}\right)\right) \\
&= w^{l+1}_{m,n} \frac{\partial}{\partial x^l_{i',j'}}\left(f\left(x^l_{i',j'}\right)\right) \\
&= w^{l+1}_{m,n} f'\left(x^l_{i',j'}\right)
\end{aligned} \tag{19}$$

CNN BackPropagation



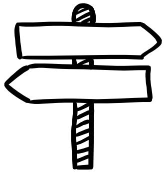
Substituting Eq. 19 in Eq. 17 gives us the following results:

$$\frac{\partial E}{\partial x^{l}_{i',j'}} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} \delta^{l+1}_{i'-m,j'-n} w^{l+1}_{m,n} f'\left(x^{l}_{i',j'}\right) \tag{20}$$

For backpropagation, we make use of the flipped kernel and as a result we will now have a convolution that is expressed as a cross-correlation with a flipped kernel:

$$\begin{aligned}
\frac{\partial E}{\partial x^{l}_{i',j'}} &= \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} \delta^{l+1}_{i'-m,j'-n} w^{l+1}_{m,n} f'\left(x^{l}_{i',j'}\right) \\
&= \mathrm{rot}_{180^\circ} \left\{ \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} \delta^{l+1}_{i'+m,j'+n} w^{l+1}_{m,n} \right\} f'\left(x^{l}_{i',j'}\right) && (21) \\
&= \delta^{l+1}_{i',j'} * \mathrm{rot}_{180^\circ} \left\{ w^{l+1}_{m,n} \right\} f'\left(x^{l}_{i',j'}\right) && (22)
\end{aligned}$$

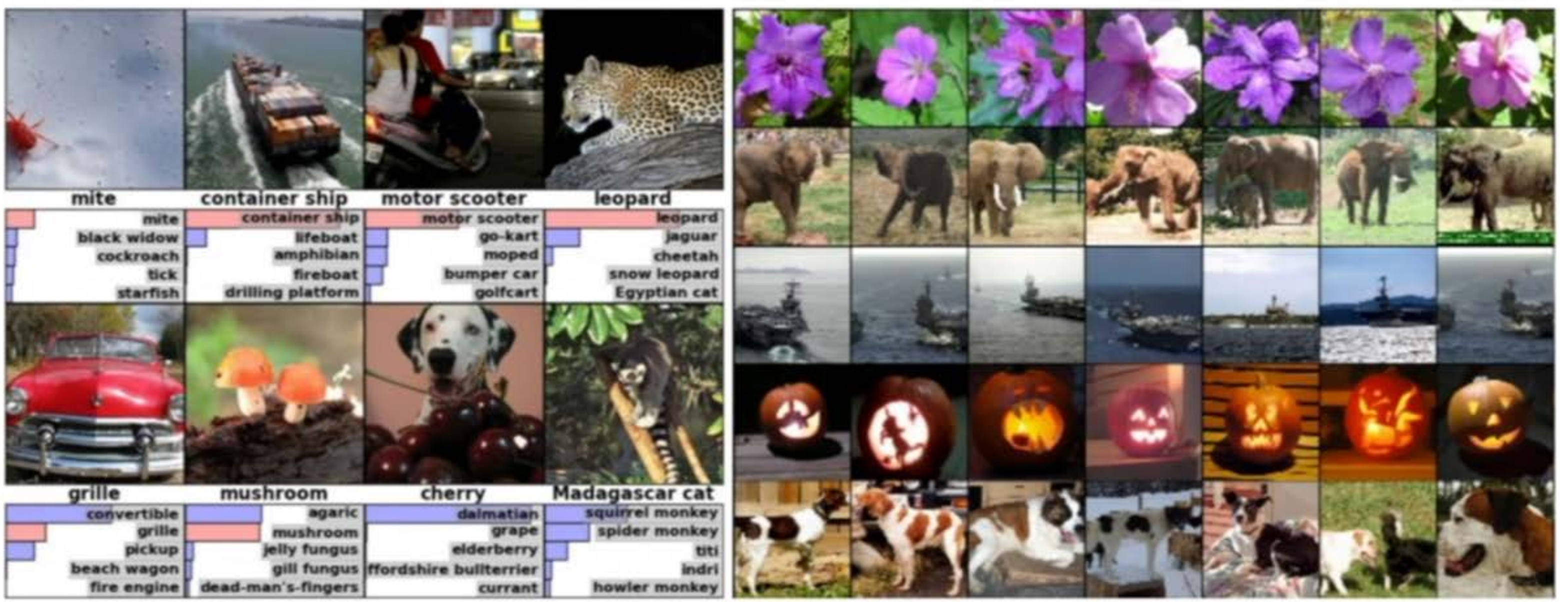
References



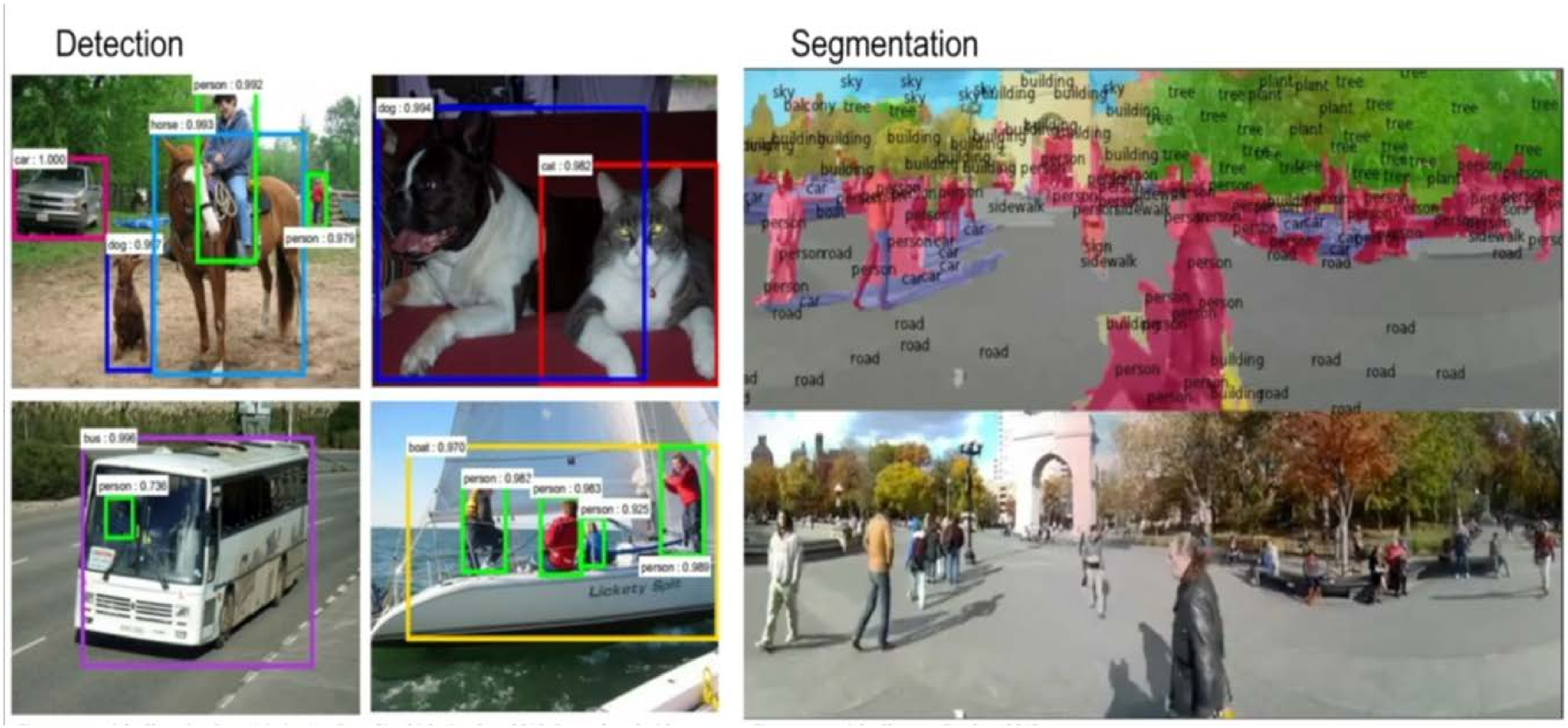
- Useful links:
  - http://courses.cs.tau.ac.il/Caffe_workshop/Bootcamp/pdf_lectures/Lecture%203%20CNN%20-%20backpropagation.pdf
  - https://medium.com/@14prakash/back-propagation-is-very-simple-who-made-it-complicated-97b794c97e5c
  - https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/

CNN Applications

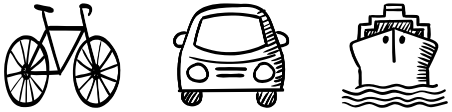
Image Classification

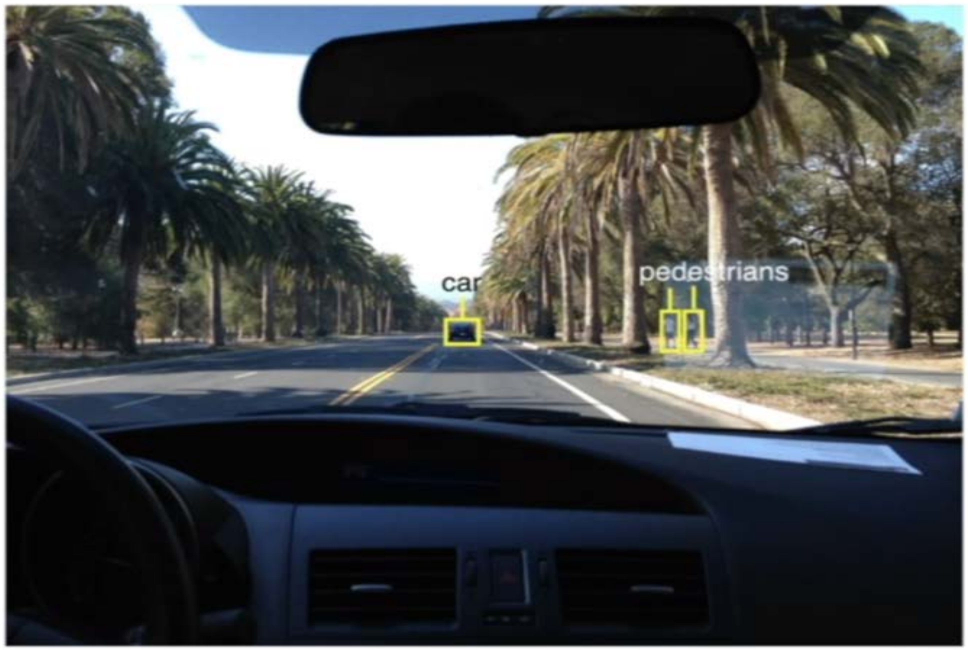


CNN Applications



CNN Applications

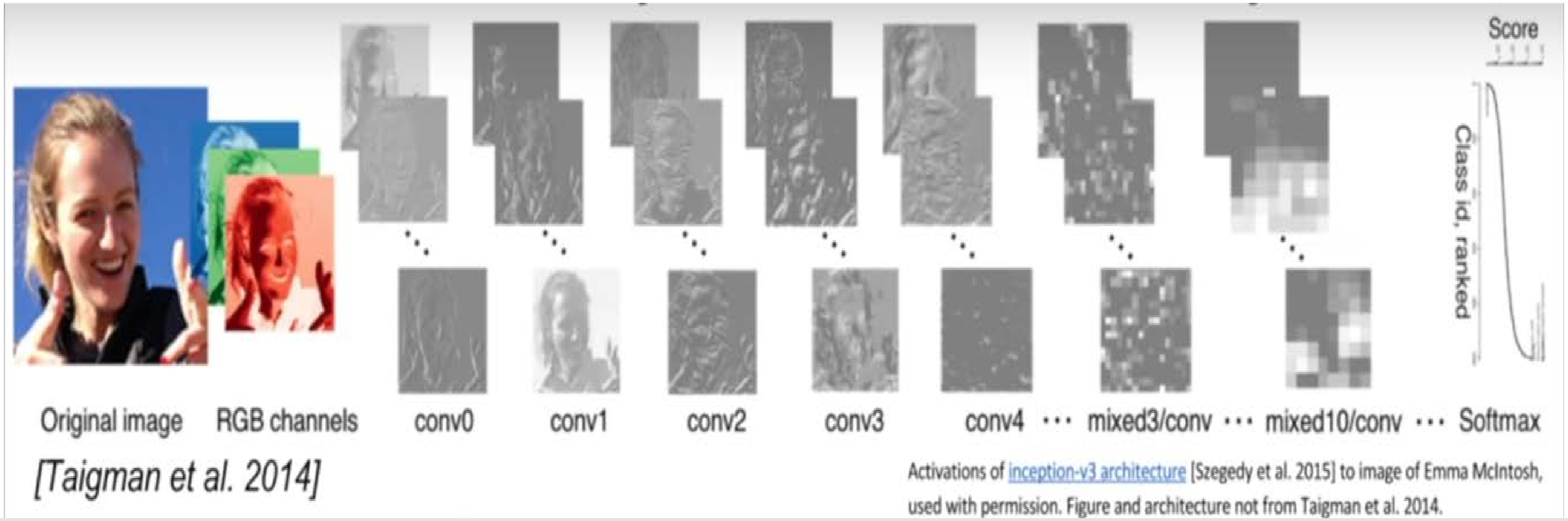


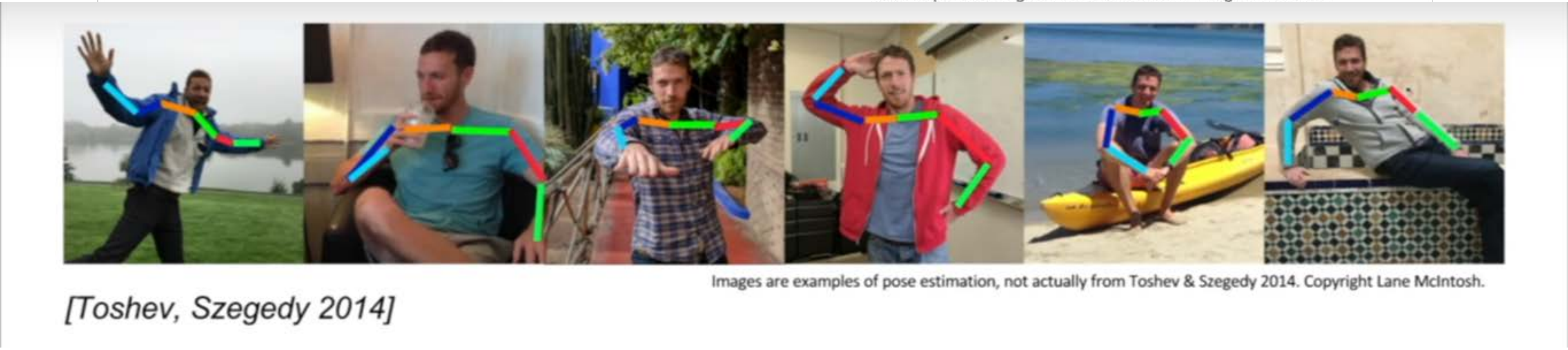


self-driving cars

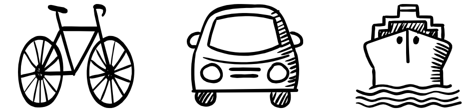
NVIDIA Tesla line
(these are the GPUs on rye01.stanford.edu)

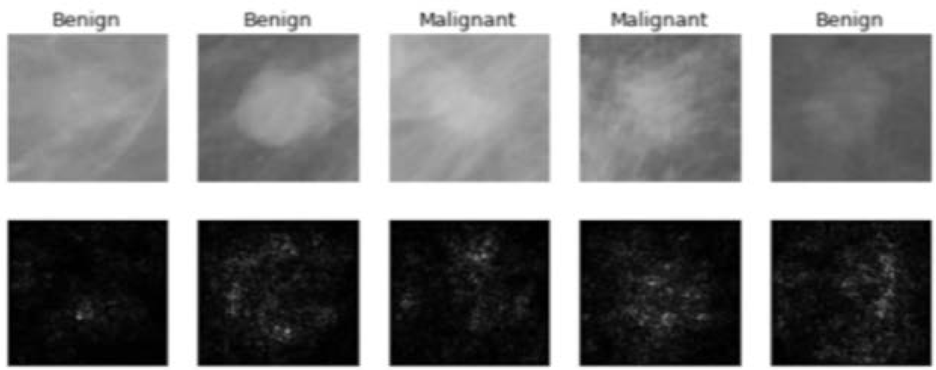
CNN Applications





CNN Applications

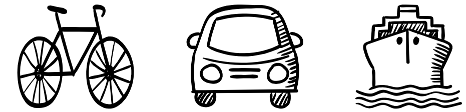




[Levy et al. 2016]

[Dieleman et al. 2014]

CNN Applications



This image by Christin Khan is in the public domain and originally came from the U.S. NOAA.

Photo and figure by Lane McIntosh; not actual example from Mnih and Hinton, 2010 paper.

Whale recognition, Kaggle Challenge

Mnih and Hinton, 2010

CNN Applications

No errors

A white teddy bear sitting in the grass

A man riding a wave on top of a surfboard

Minor errors

A man in a baseball uniform throwing a ball

A cat sitting on a suitcase on the floor

Somewhat related

A woman is holding a cat in her hand

A woman standing on a beach holding a surfboard

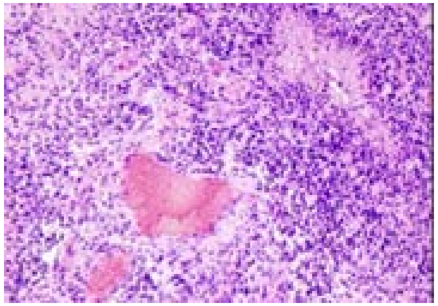
Image Captioning

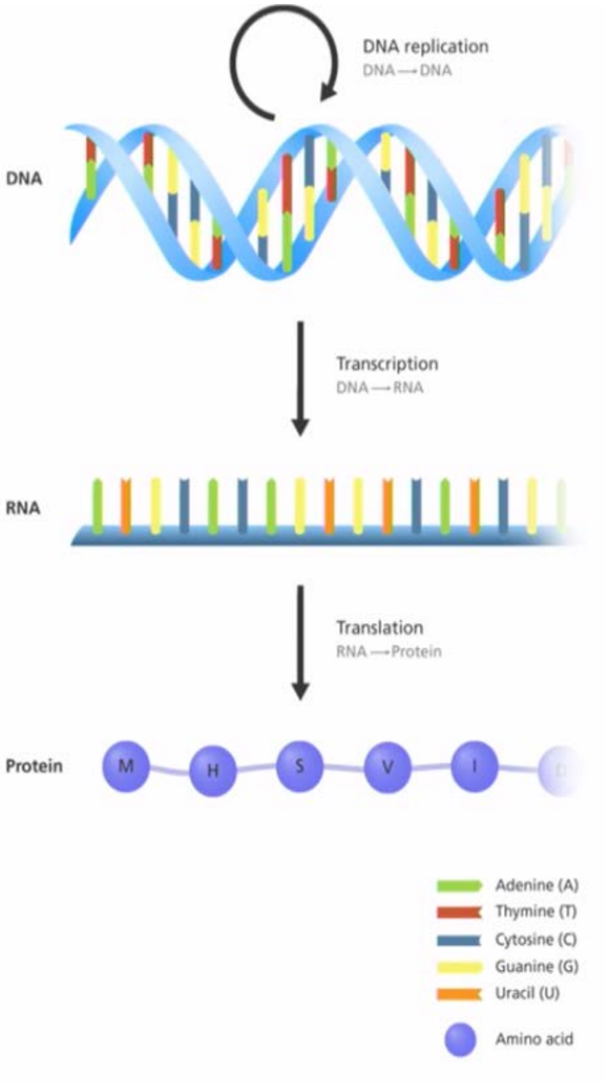
[Vinyals et al., 2015] [Karpathy and Fei-Fei, 2015]

Captions generated by Justin Johnson using Neuraltalk2

CNN in Medical Imaging

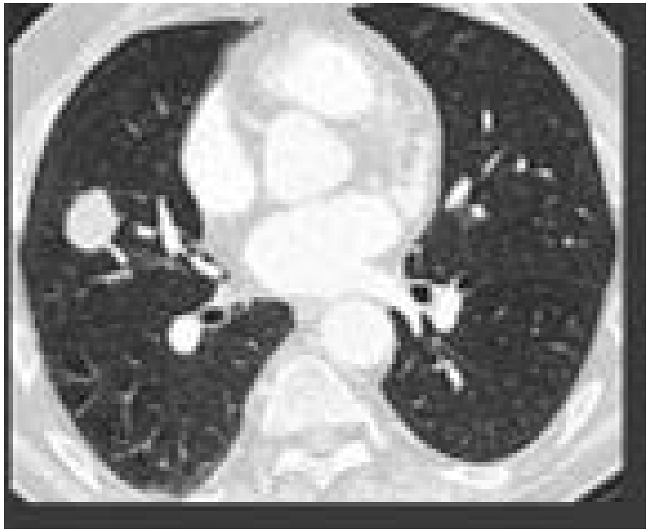
- Biomedical Data:
  - Examples of biomedical data:
    - The omics: Genome, Transcriptome
    - Imaging: Pathology, Radiology





CNN in Medical Imaging

- Volumetric data => 3D images
- MRI is multimodal
- Complementary information
    - Size, location, morphology



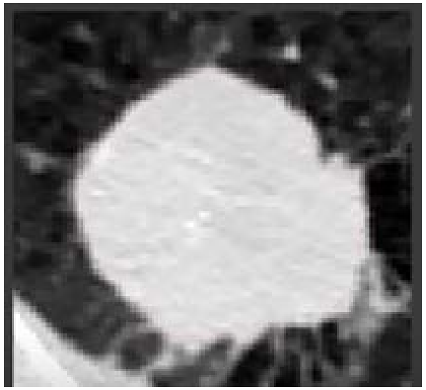
CNN in Medical Imaging

- Two types of image features
  - Semantic features
    - Image features manually annotated by radiologist
    - Radiologist-dependent interpretation of an image
    - Most likely qualitative
  - Computational features
    - (semi)-automatically extracted from an image
    - Most likely quantitative

CNN in Medical Imaging

- Semantic features:
  - Edge shape

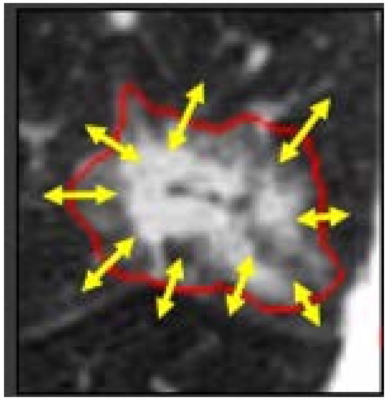
Smooth



Lobulated

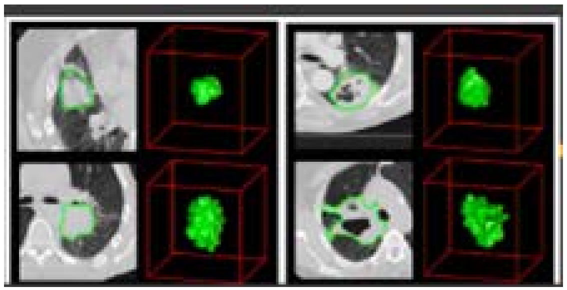
CNN in Medical Imaging

- Computational Features:
  - Texture Features
  - Shape Features
  - Edge Sharpness Features



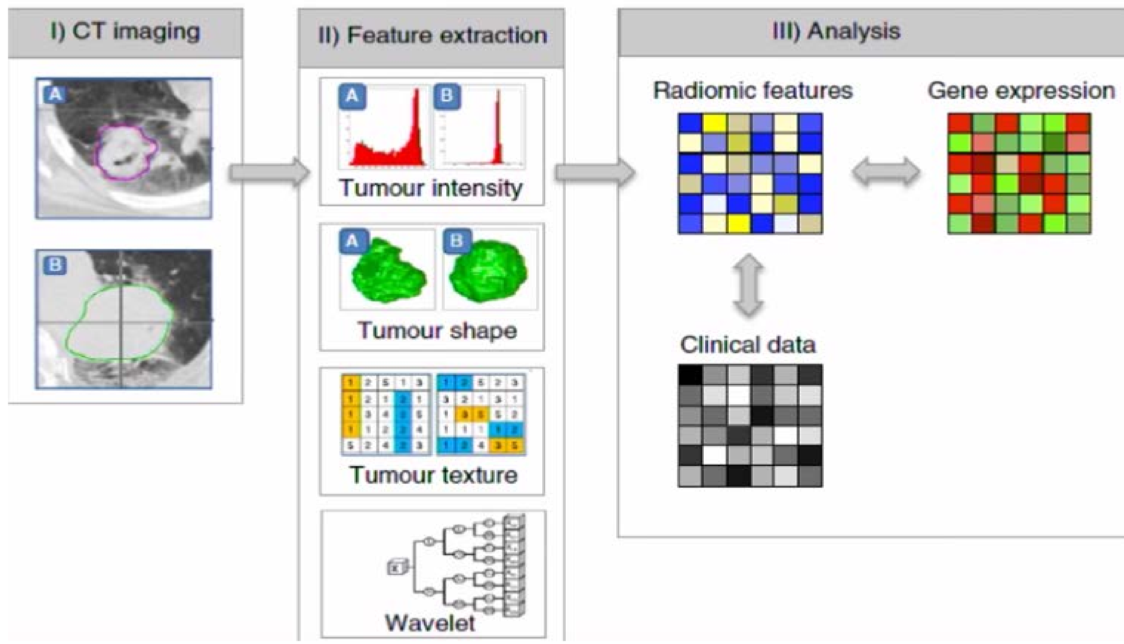
CNN in Medical Imaging

Given CT scan of a patient can we predict survival?

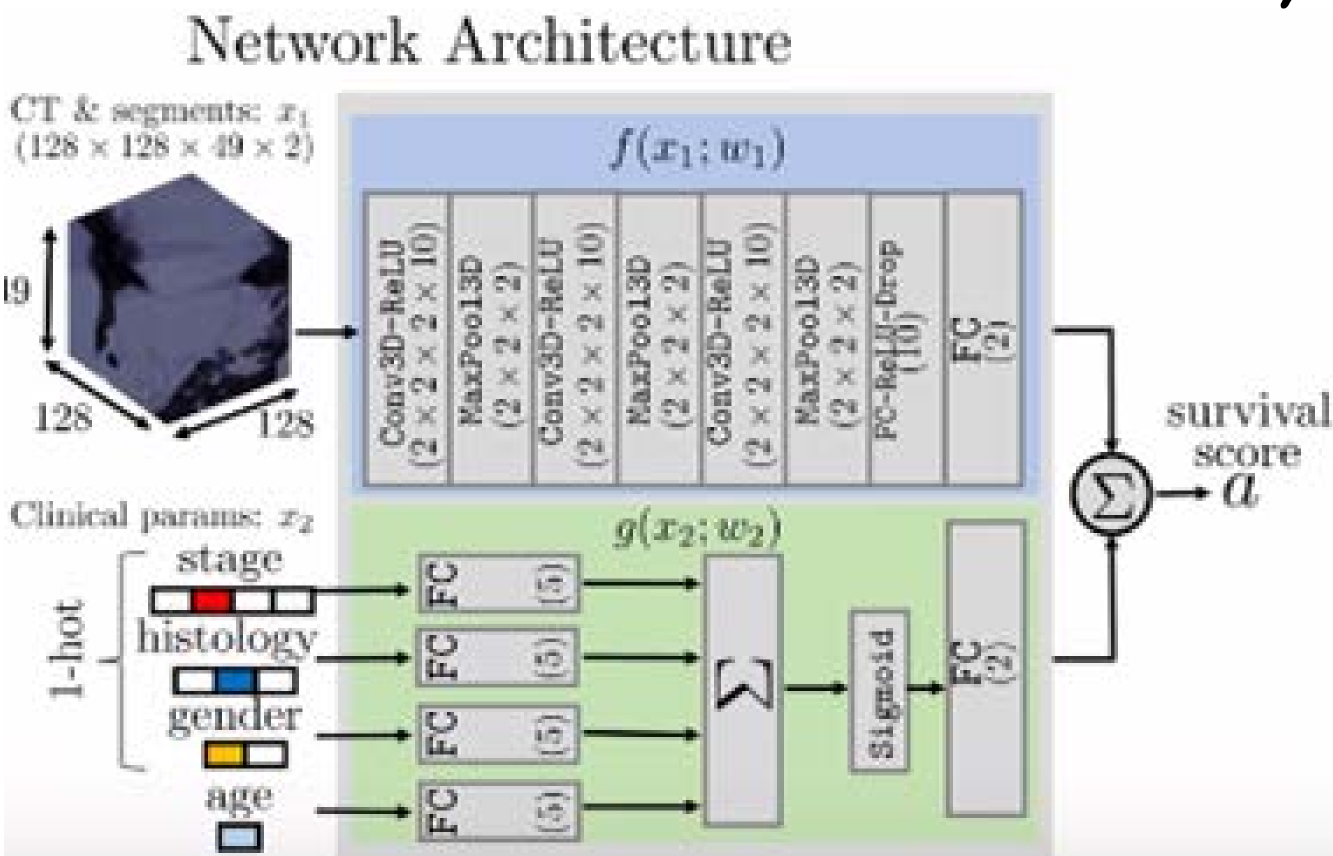


Survival score?

Conventional Approach

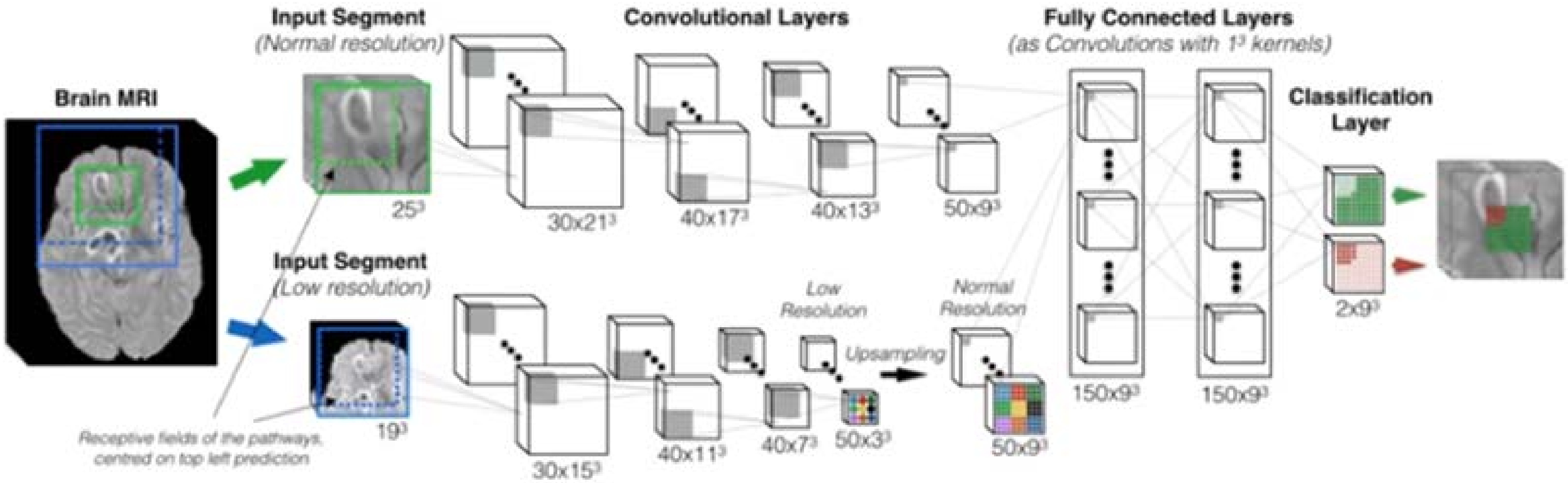


CNN in Medical Imaging



CNN in Medical Imaging

brain lesions



[Kamnitsas et al. 2016]

CNN in Medical Imaging