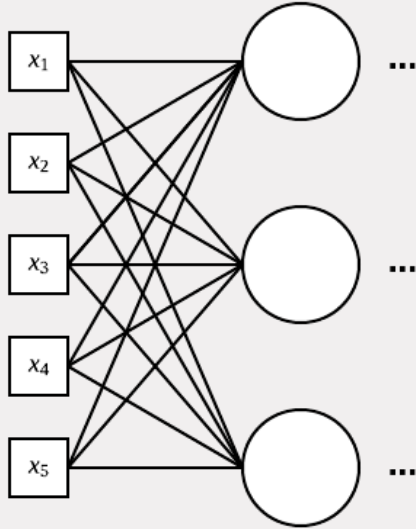


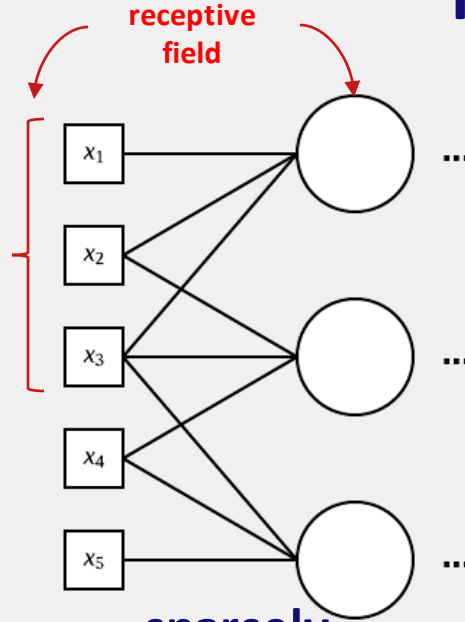
Deep Learning Frameworks and Applications (8DC00)

Navchetan Awasthi

Previous lecture

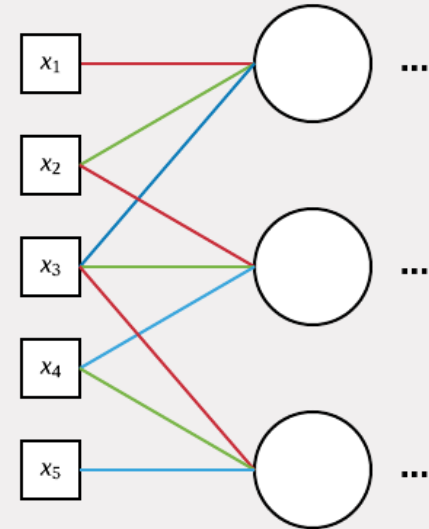


“regular” NN
15 weights



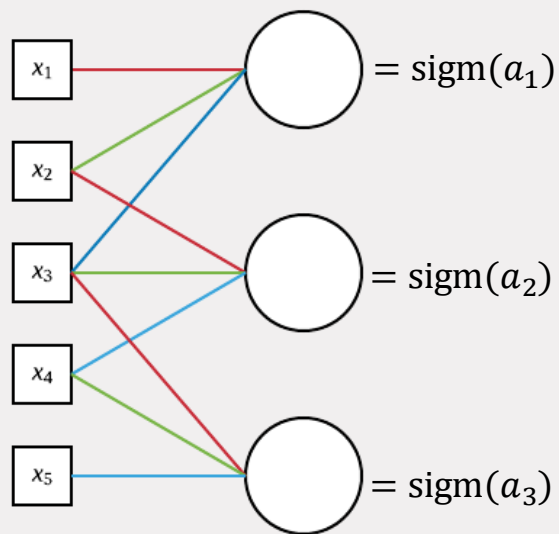
**sparsely
connected NN**
9 weights

Reducing # of weights



shared weights
3 weights

Previous lecture



$$a_1 = x_1 w_1 + x_2 w_2 + x_3 w_3$$

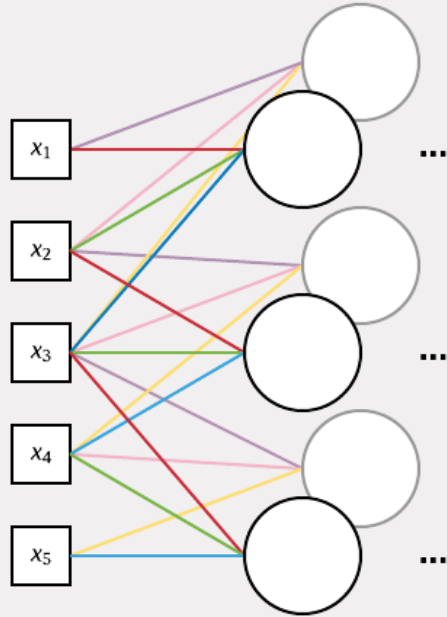
$$a_2 = x_2 w_1 + x_3 w_2 + x_4 w_3$$

$$a_3 = x_3 w_1 + x_4 w_2 + x_5 w_3$$

$$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix} * \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix}$$

convolution

Previous lecture



two sets shared weights
6 weights

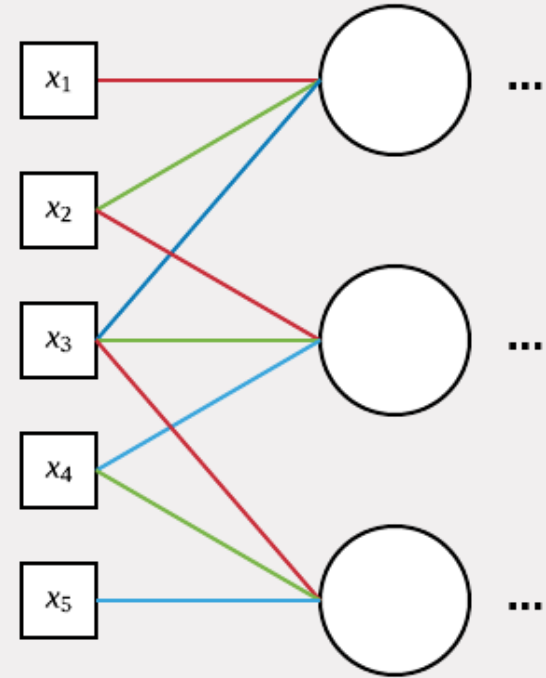
$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix} * \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \end{bmatrix}$$

$$\begin{bmatrix} a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix} * \begin{bmatrix} w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix}$$

$\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \end{bmatrix}$, and $\begin{bmatrix} w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix}$
are **convolution kernels**. They extract features.

Previous lecture: properties of CNNs

- Sparse connectivity
- Weight sharing
- Parallel computations

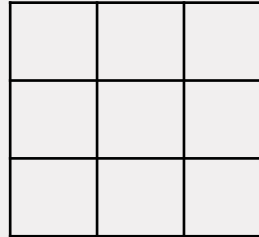


Previous lecture: Kernel size

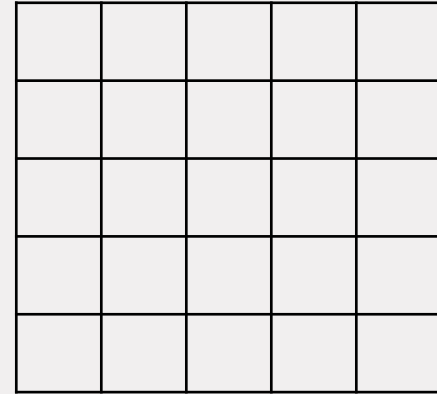
1 x 1



3 x 3



5 x 5



- More weights → more information
- More computations/ memory
- Receptive field

Previous lecture: Max-pooling

- Reduce size of feature space
- Maximum of features
- Typical kernel size = 2×2

Addition to last lecture:

Stride = step size of the sliding window

- Typically 1 for convolutions
- Typically equal to kernel size for pooling operations

Draw your number here

0123456789

3



Downsampled drawing: 3

First guess: 3

Second guess: 1

Layer visibility

Input layer

Show

Evolution layer 1

Show

Downsampling layer 1

Show

Evolution layer 2

Show



Previous lecture: Training of CNNs

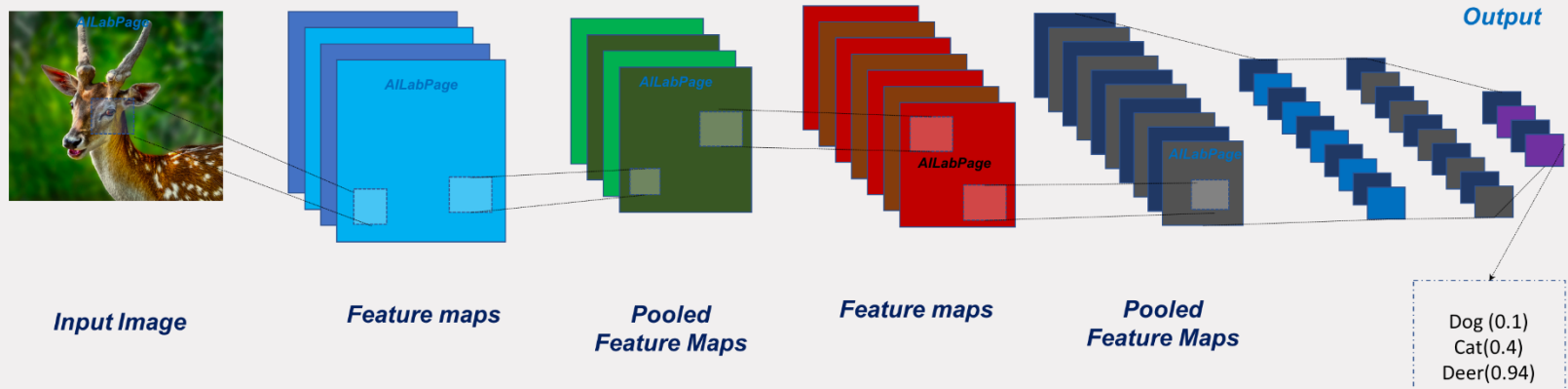
- Similar to training of 'fully connected' neural networks
- **Choose some (random) initial values for network weights**
- Optimize networks weights with respect to a loss function that describes difference between network output and label/annotation
- Update networks weights iteratively

Through a process called 'backpropagation'. A good explanation can be found here:
<https://www.youtube.com/watch?v=i94OvYb6n0o> (5:10 - 28:00, not exam material)

- Keep track of model performance on train & validation set

Previous lecture

- Convolutional neural networks (CNN)



- By now: students can design a (simple) CNN

Image from: <https://vinodsblog.com/>

Challenges..

Disadvantage of designing yourself..?

- New implementation
- Design choices
- Time-consuming
- Debugging
- No reference performance
- Non-optimized model
- Train from scratch

Deep Learning Framework

Model architecture (from literature)
+
Framework (software packages)

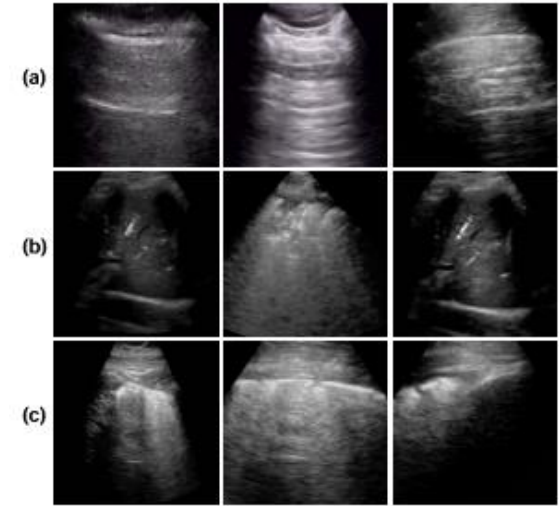
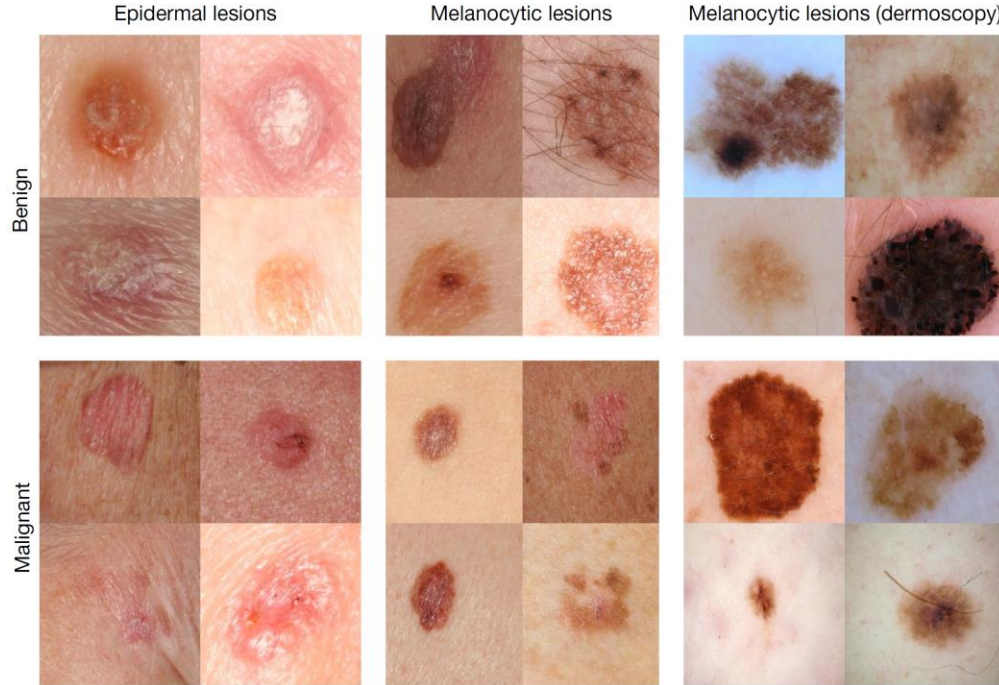
Learning outcomes

- Students will be able to name a few **deep learning models** and describe the **differences in model design**
- Students can motivate why they **would choose a deep learning model** for a specific research application
- Students can name **two key developments** that have enabled deep learning

Lecture outline

- Different applications → Different requirements
- Small exercise: “what should be the output?”
- History of deep learning models
- Examples of deep learning models & applications
- *Break (15 mins)*
- Examples of deep learning software
- Group exercise: “*choosing the right deep learning approach*”

Classification



Example of lung US images utilized in this work representing classes (rowwise) of (a) healthy lung, (b) pneumonia infected lung, and (c) lung infected with COVID-19 exhibiting pleural irregularities and small subpleural consolidation.

Esteva et al.(Nature 2017), Awasthi et al. (TUFFC 2021)

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{1 + \sum_j e^{z_j}}$$

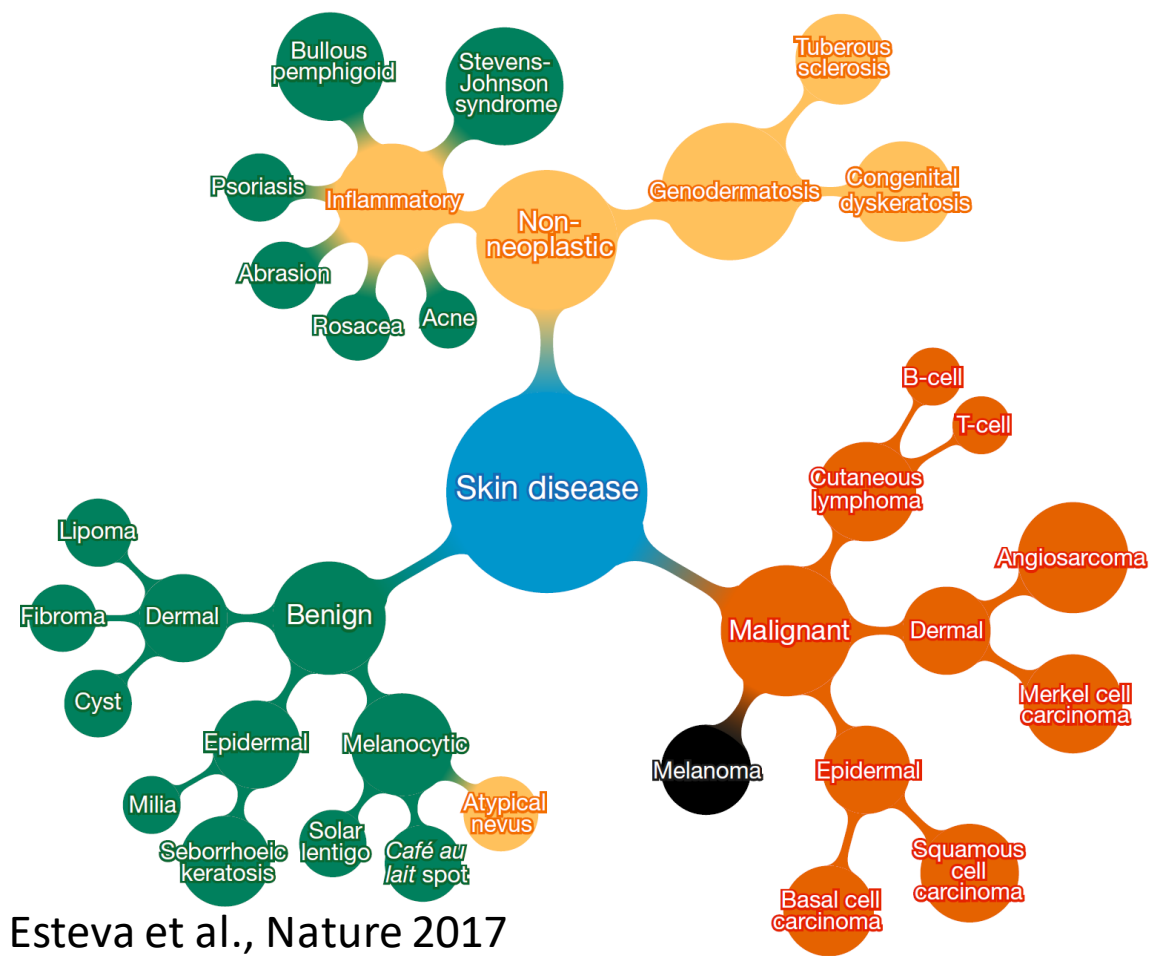
classification

Two classes

e.g. malignant lesions / other

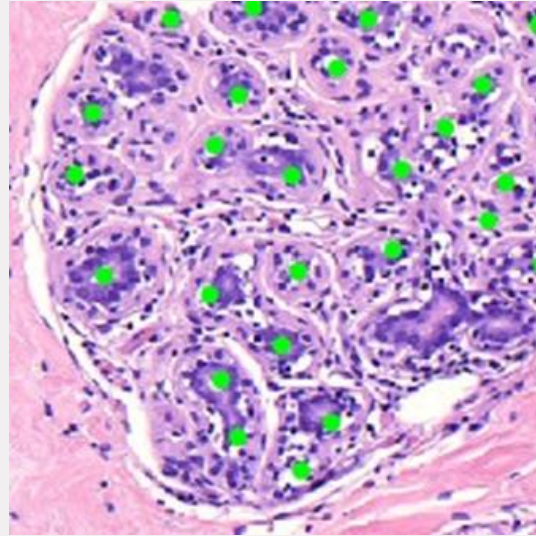
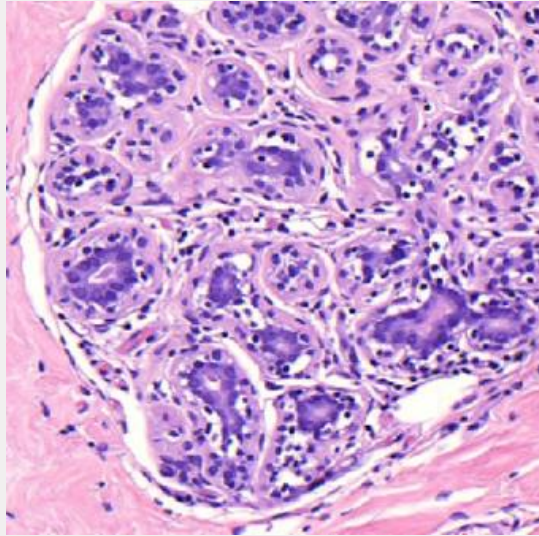
Multi-class

benign lesions / malignant lesions
/ non-neoplastic



Esteva et al., Nature 2017

Detection of Acini centers



Wetstein et al., SPIE-MI 2019

Regression

Retinal structures

- Arteriolar width
- Venular width
- Vessel Tortuosity

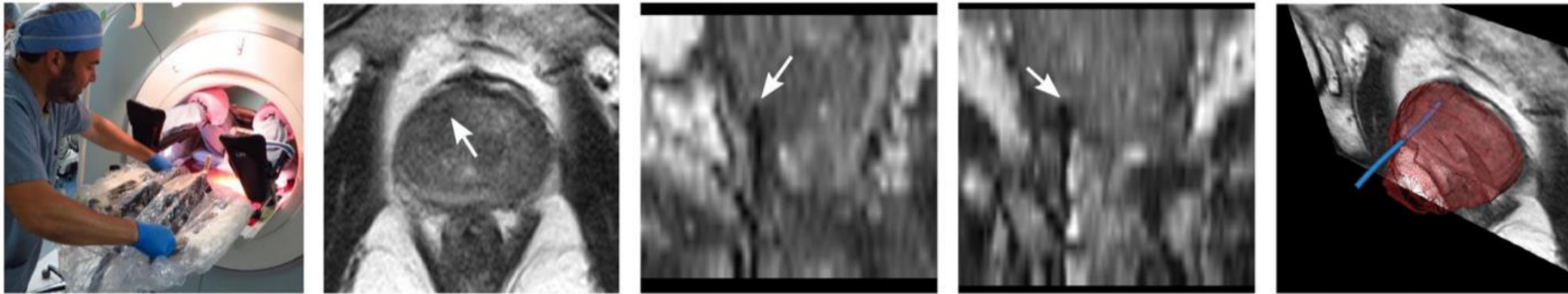
Heslinga et al., SPIE-MI 2019



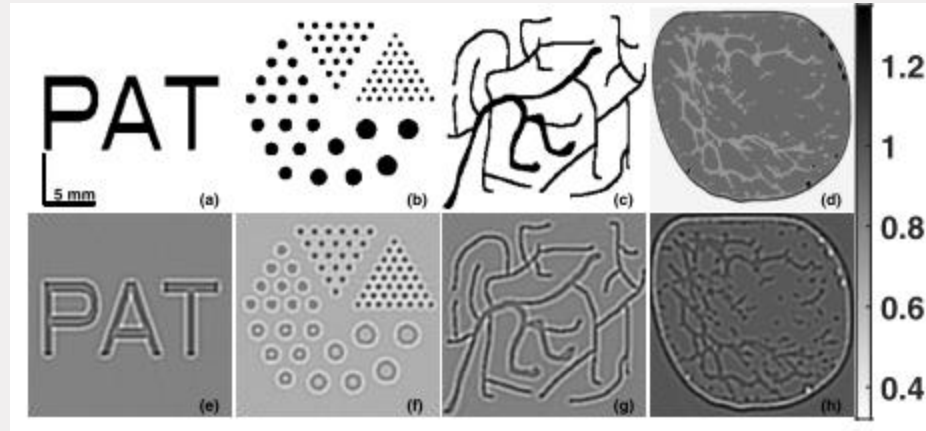
Segmentation

- MRI-targeted prostate biopsy
- Needle segmentation

Mehrtash et al., IEEE Trans Med Imaging 2019



Reconstruction



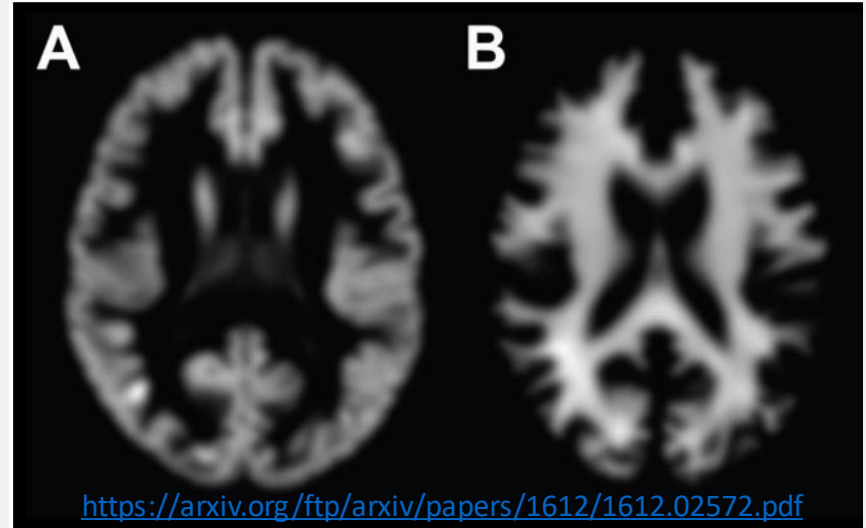
Target numerical phantoms used for quantitative evaluation of the proposed method. (a)–(d) Ground-truth images for PAT, modified Derenzo, BV, and breast phantoms. Corresponding reconstructed images using LBP with full BW and noise less data are shown in (e)–(h) to serve as reference for best case.

Awasthi et al. (TUFFC 2019)

What should be the output of our CNN?

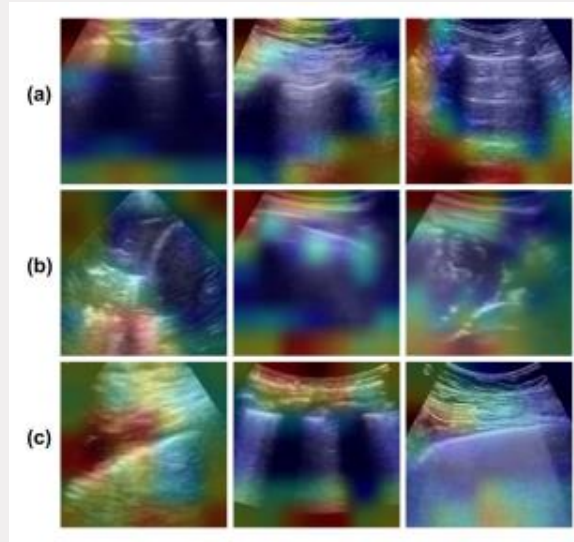


Retinopathy classification



Brain age prediction

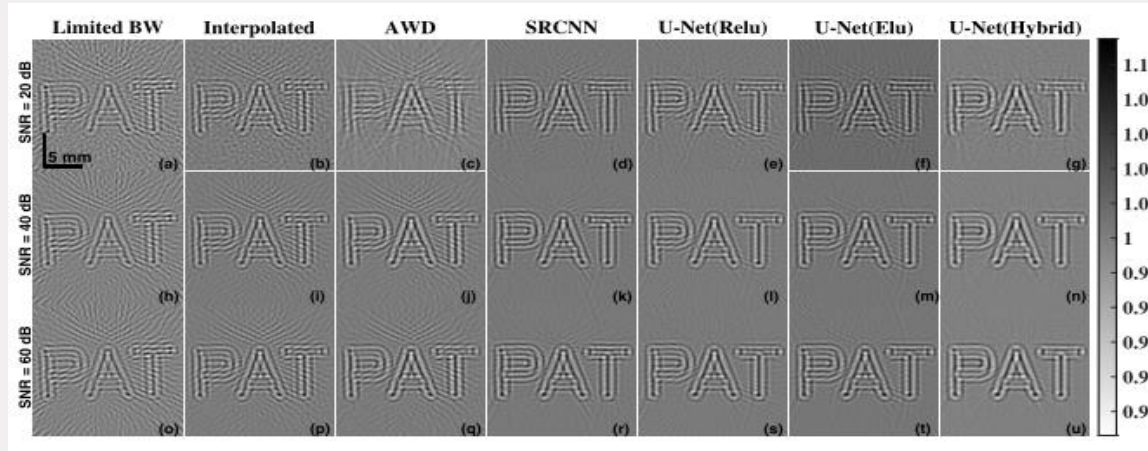
What should be the output of our CNN?



Example lung US images after the visualization using the Grad-CAM utilized in this work representing classes (rowwise) of (a) healthy lung, (b) pneumonia infected lung, and (c) lung infected with COVID-19 exhibiting pleural irregularities and small subpleural consolidation.

Awasthi et al. (TUFFC 2021)

What should be the output of our CNN?

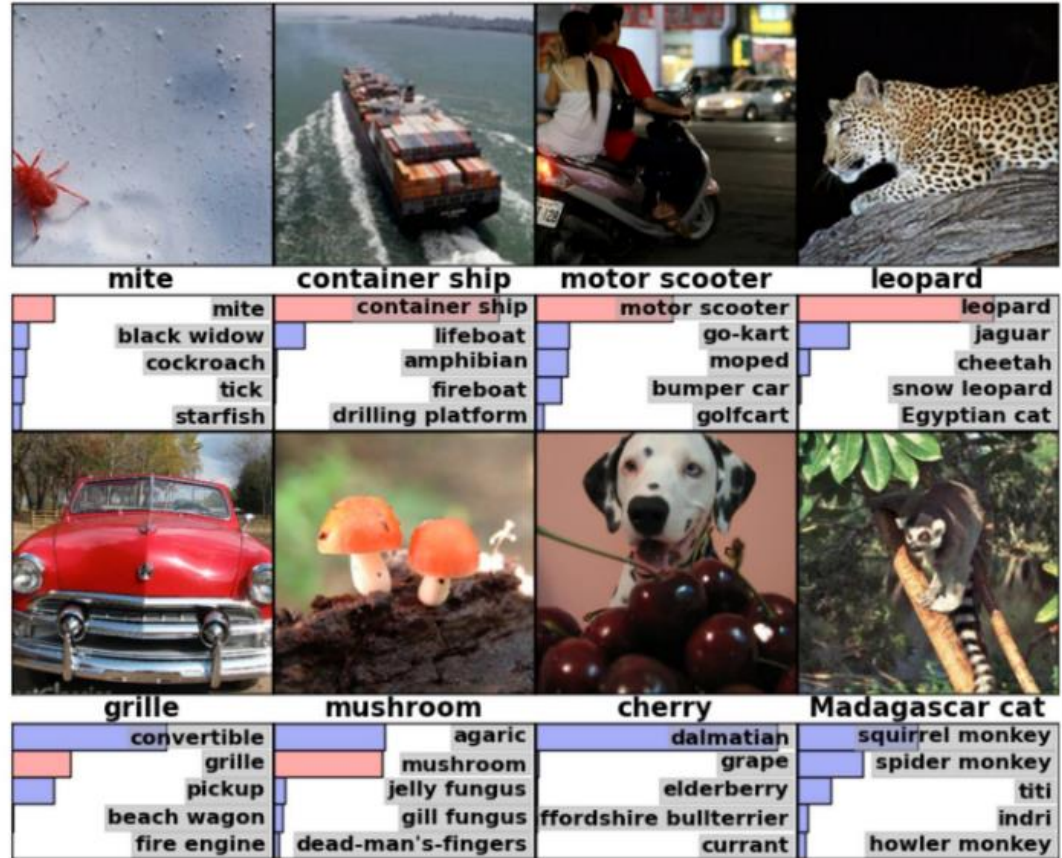


Reconstructed PAT phantom images with each row representing the SNR of data being 20, 40, and 60 dB and correspondingly given in (a)–(g), (h)–(n), and (o)–(u), respectively. Reconstructed images using data with limited BW (70%) and 100 transducers are shown in (a), (h), and (o). While (b), (i), and (p) show reconstruction results when the data with 100 transducers that are interpolated using nearest neighbor to obtain a 200 transducer data. (c), (j), and (q) Results obtained using the automated wavelet denoising approach. The results obtained using the SRCNN-based method are shown in (d), (k), and (r), and the results obtained using the U-Net (Relu) are shown in (e), (l), and (s). Finally, (f), (m), and (t) show the results obtained using the proposed U-Net (Elu) approach, and (g), (n), and (u) show the results obtained using the proposed U-Net (Hybrid)-based approach. Table I reports the figure of merits of these results in a comparative manner.

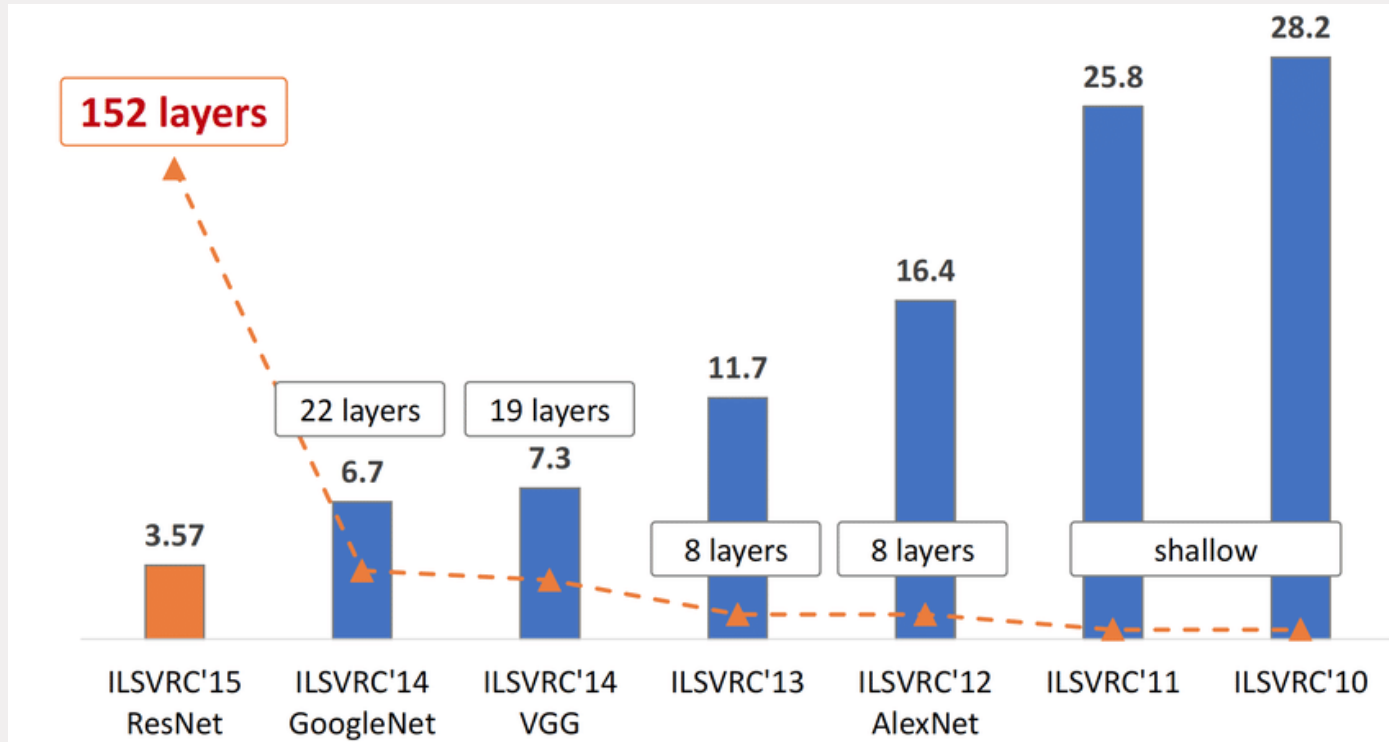
Awasthi et al. (TUFFC 2019)

IMAGENET

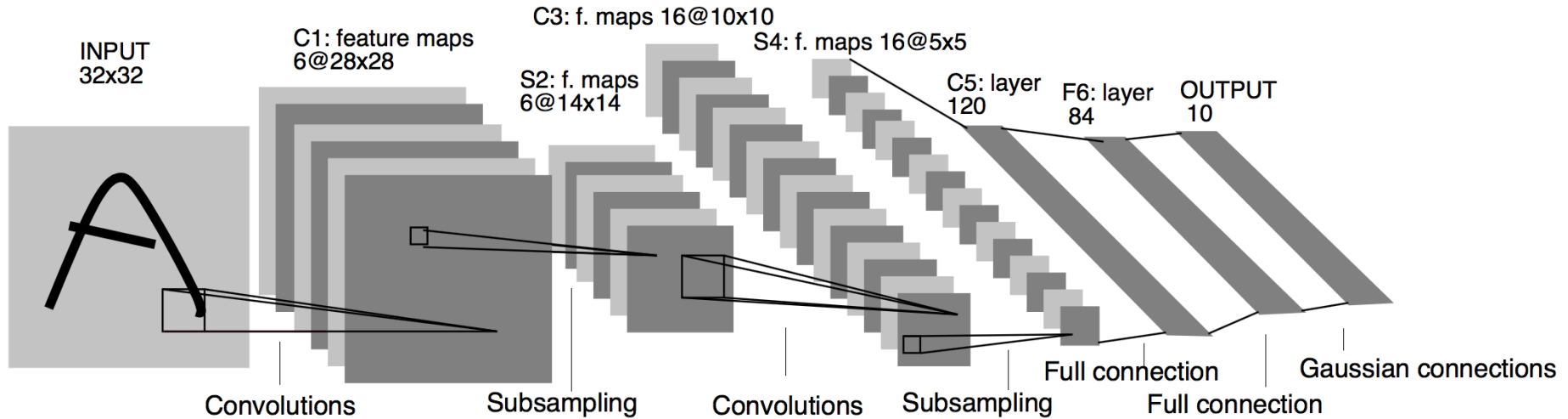
- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.



Error rates ImageNet

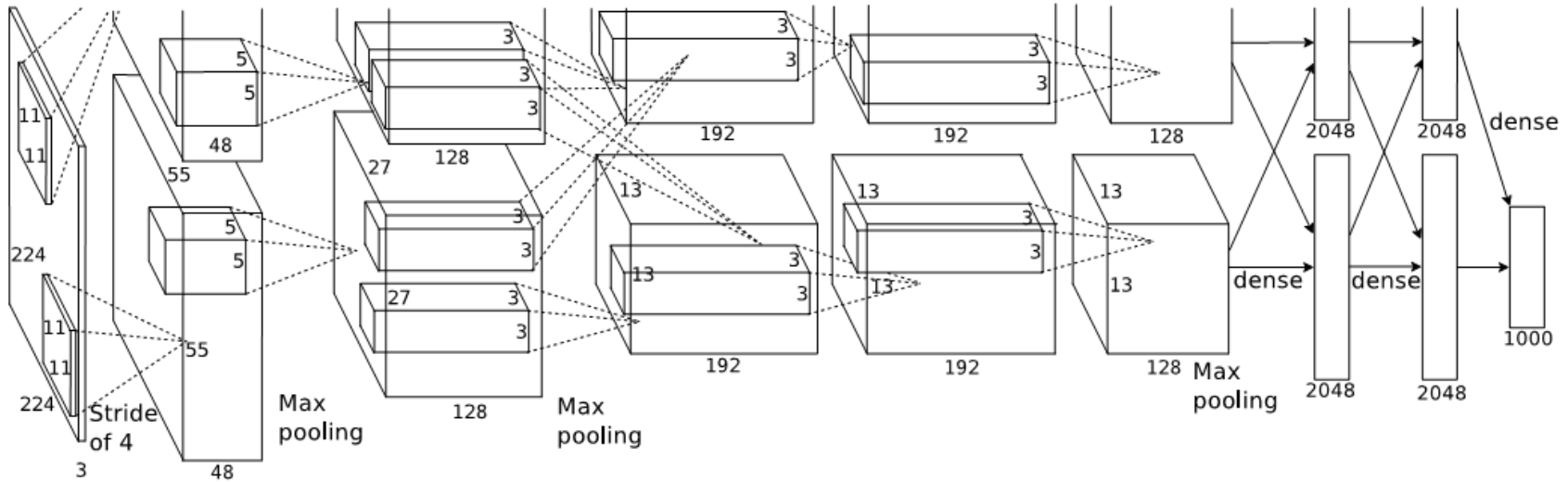


LeNet



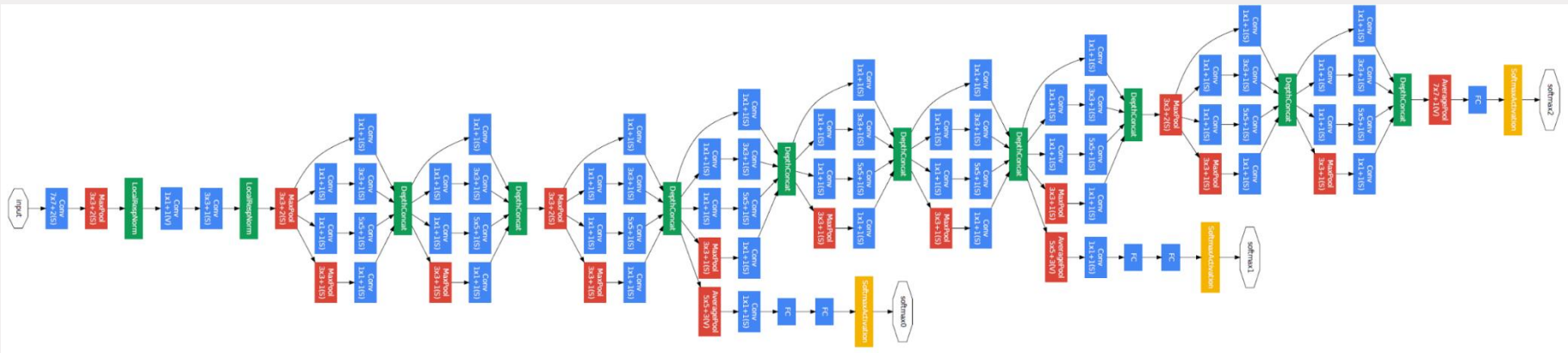
AlexNet

- ImageNet top-5 error from 26% to 15.3%



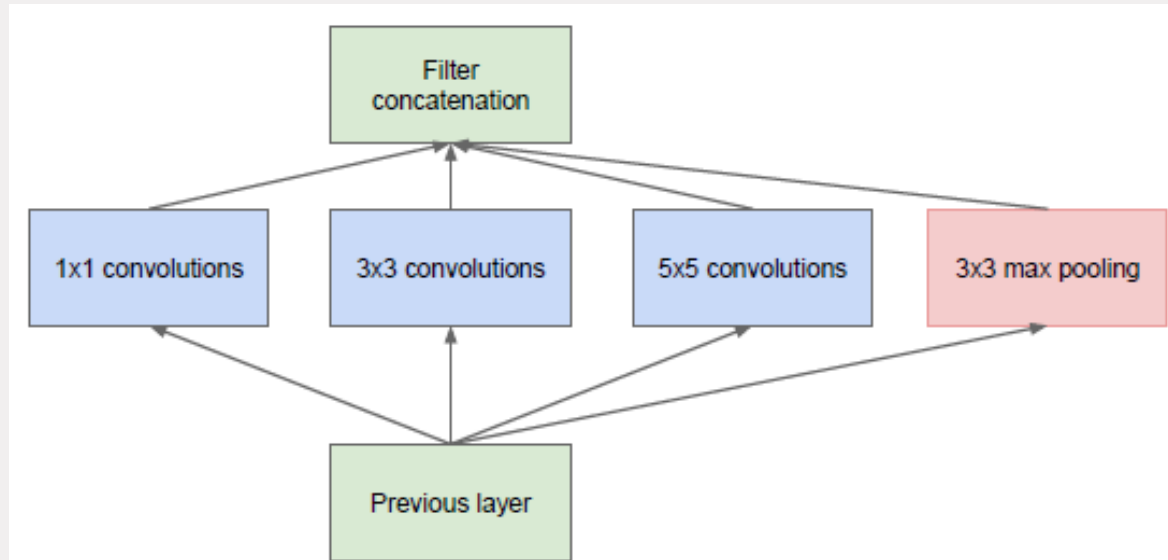
Inception v1 (GoogLeNet)

- Inception modules
- Auxiliary targets
- Global average pooling layer

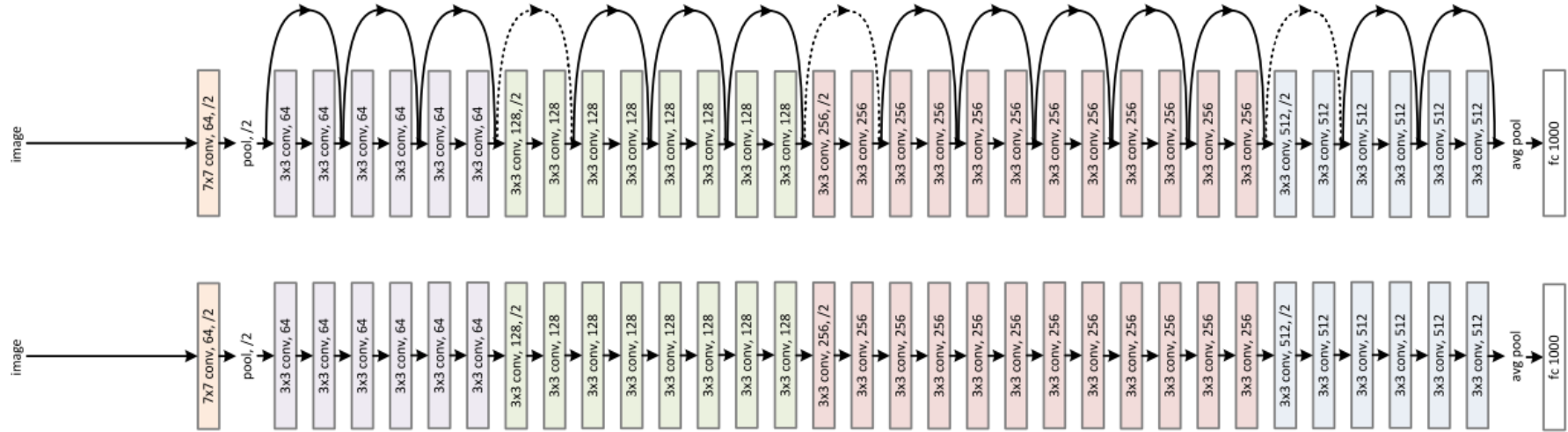
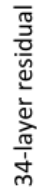


Inception module

- Multiple scales

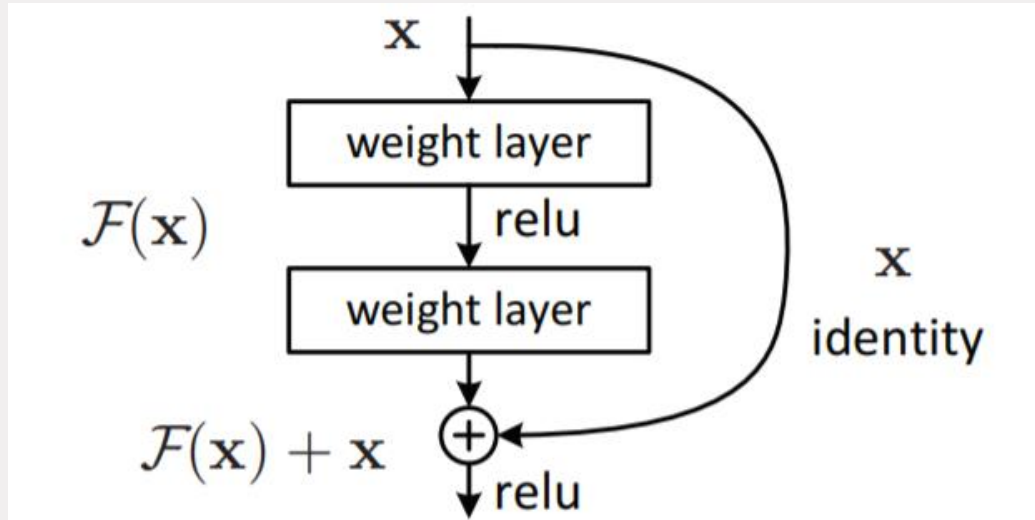


ResNet

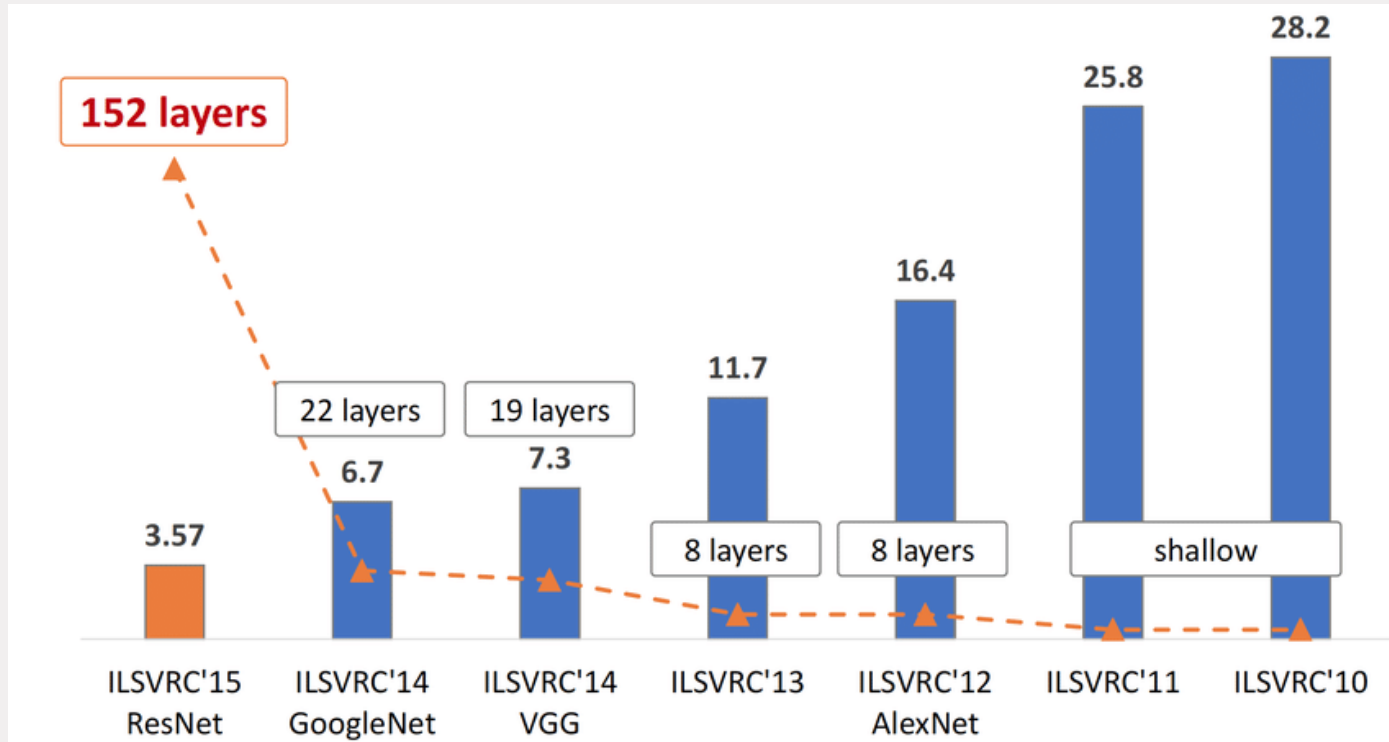


Residual block

Learn differences between input and output layer = residue

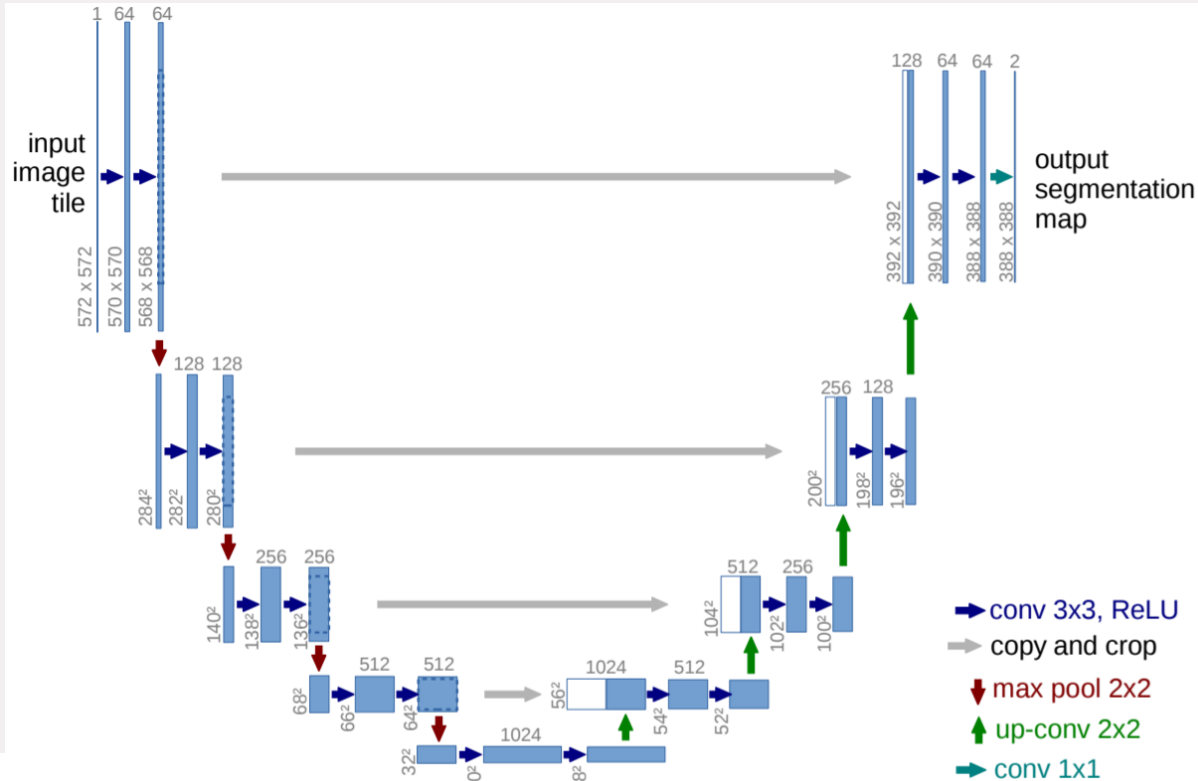


Error rates ImageNet (top 5 accuracy)



U-Net: Segmentation

Ronneberger et al., MICCAI 2015



Neural networks have been around

LEARNING INTERNAL REPRESENTATIONS BY ERROR PROPAGATION

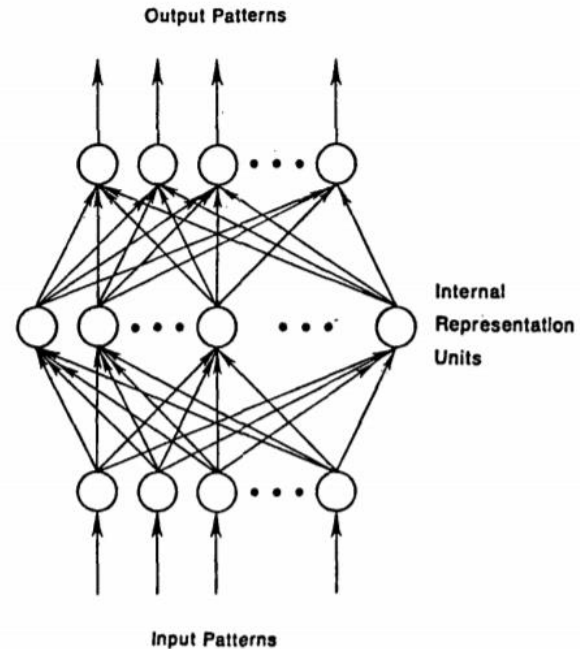
David E. Rumelhart, Geoffrey E. Hinton,
and Ronald J. Williams

September 1985

ICS Report 8506

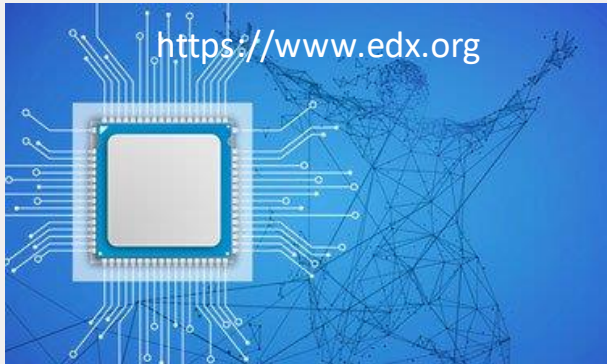


COGNITIVE
SCIENCE



Deep learning enabled

Why did it take so long to develop these deep learning models?



- Hardware improvements
- Parallel computing (GPU)



- Digital data
 - More (image) data
- = also true for medical imaging

Lecture outline

- Different applications → Different requirements
- Small exercise: “what should be the output?”
- History of deep learning models
- **Examples of deep learning models & applications**
- *Break (15 mins)*
- Examples of deep learning software
- Group exercise: *“choosing the right deep learning approach”*

Example 1: Skin lesion classification

129,450 clinical images

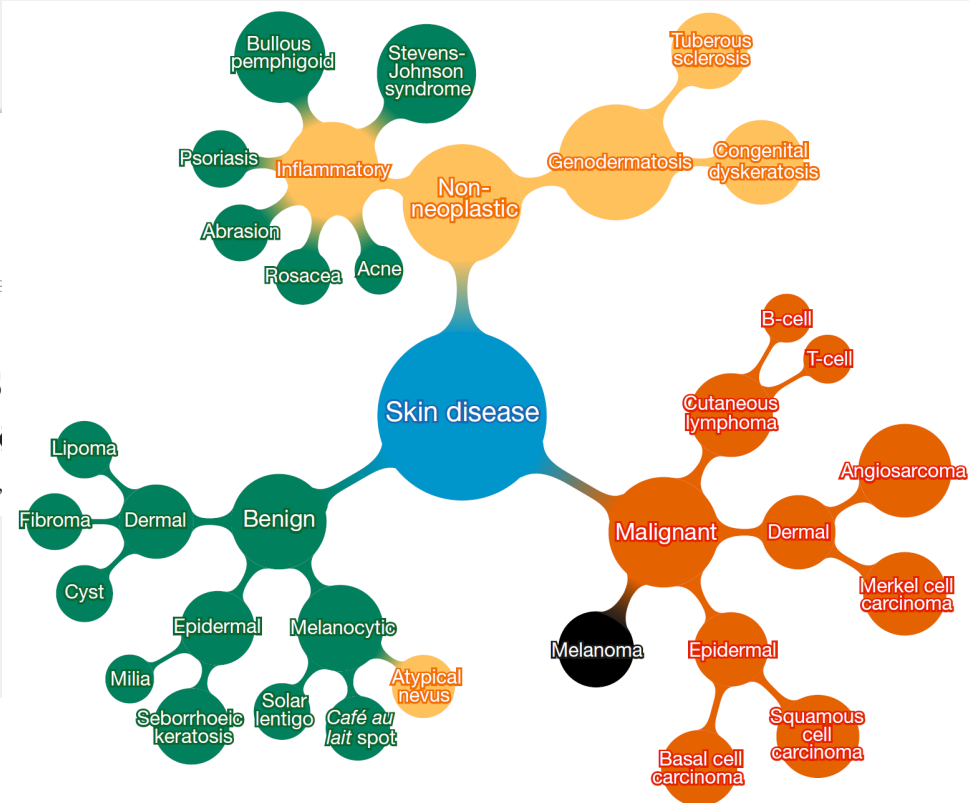
2,032 different diseases

3 classes

LETTER

Dermatologist-level class with deep neural network

Andre Esteva^{1*}, Brett Kuprel^{1*}, Roberto A. Novoa^{2,3}, Justin Ko²,

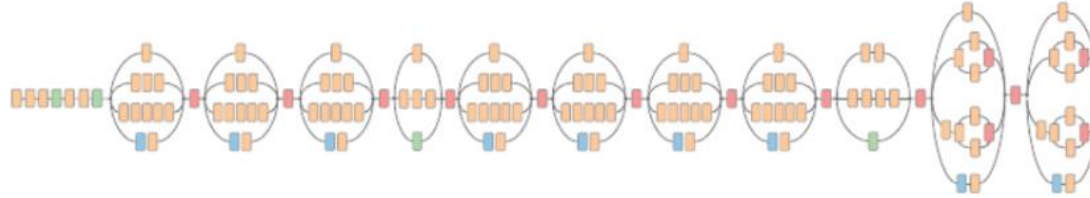


Inception v3

Skin lesion image

Deep convolutional neural network (Inception v3)

Training classes (757)



- Convolution
- AvgPool
- MaxPool
- Concat
- Dropout
- Fully connected
- Softmax

- 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



Dermatologist-level?

- 3 class classification → $72.1 \pm 0.9\%$ (mean \pm s.d.) accuracy
- Two dermatologist → 65.56% and 66.0% accuracy
- But, only for 1 task!

LETTER

doi:10.1038/nature21056

Dermatologist-level classification of skin cancer with deep neural networks

Andre Esteva^{1*}, Brett Kuprel^{1*}, Roberto A. Novoa^{2,3}, Justin Ko², Susan M. Swetter^{2,4}, Helen M. Blau⁵ & Sebastian Thrun⁶

Example 2: Retinal vessel geometry approximation

- Color fundus images
- Vessel geometry related to systemic diseases (e.g. Type 2 Diabetes, hypertension)
- Approximate geometries measured with (slow) classic image analysis tools

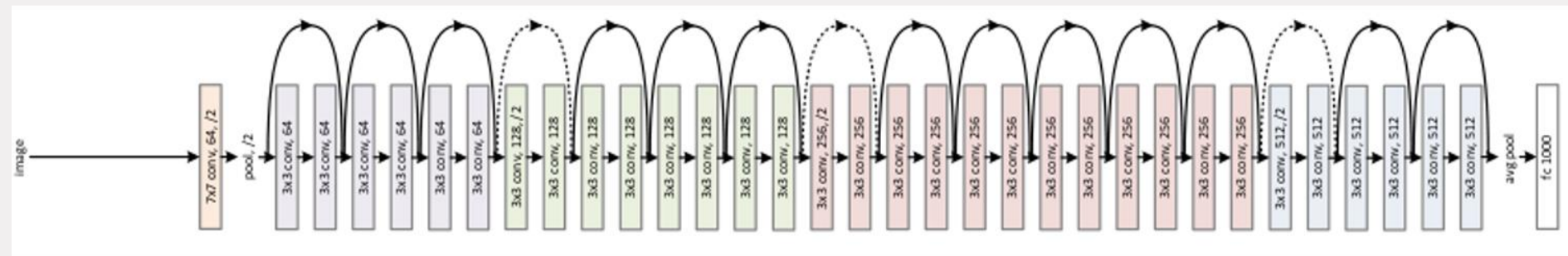
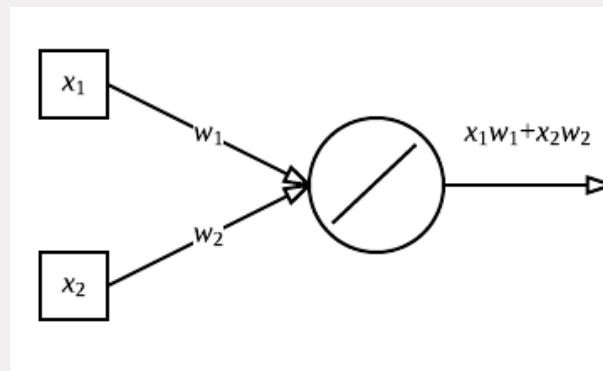
Biomarker	Description
Arteriolar width	Average width of arterioles within 0.5 to 1.0 disc diameters from the optic disc border
Venular width	Average width of venules
[...]	



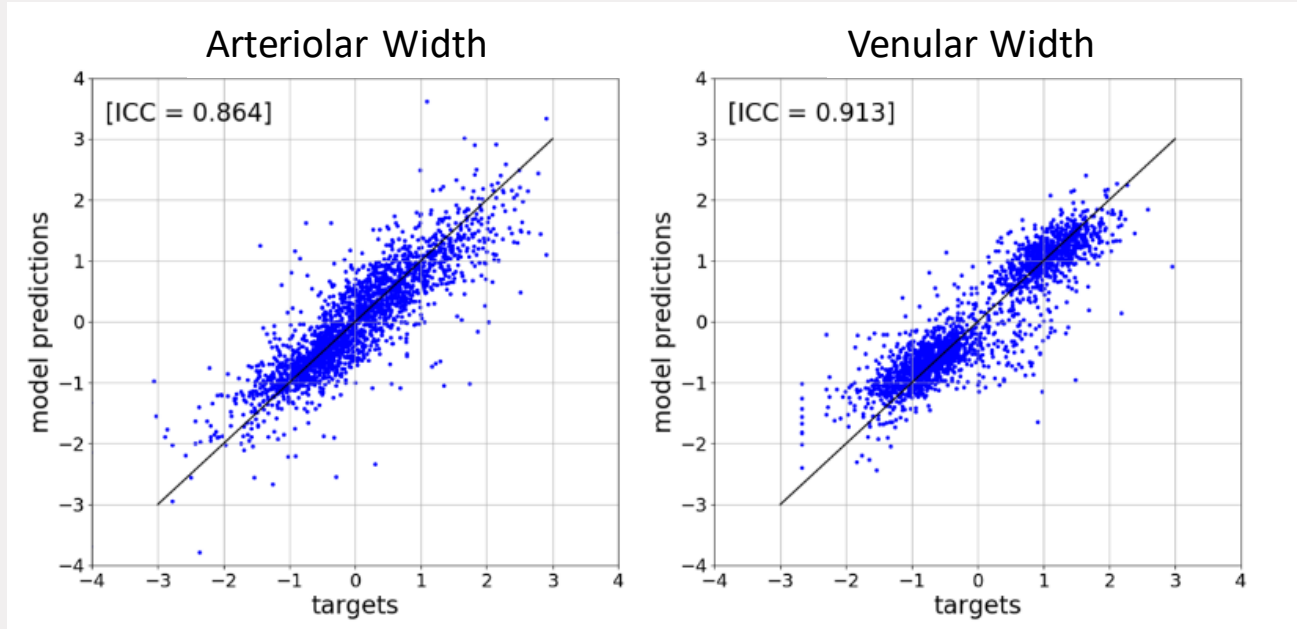
$$\hat{y} = \theta^T \mathbf{x}$$

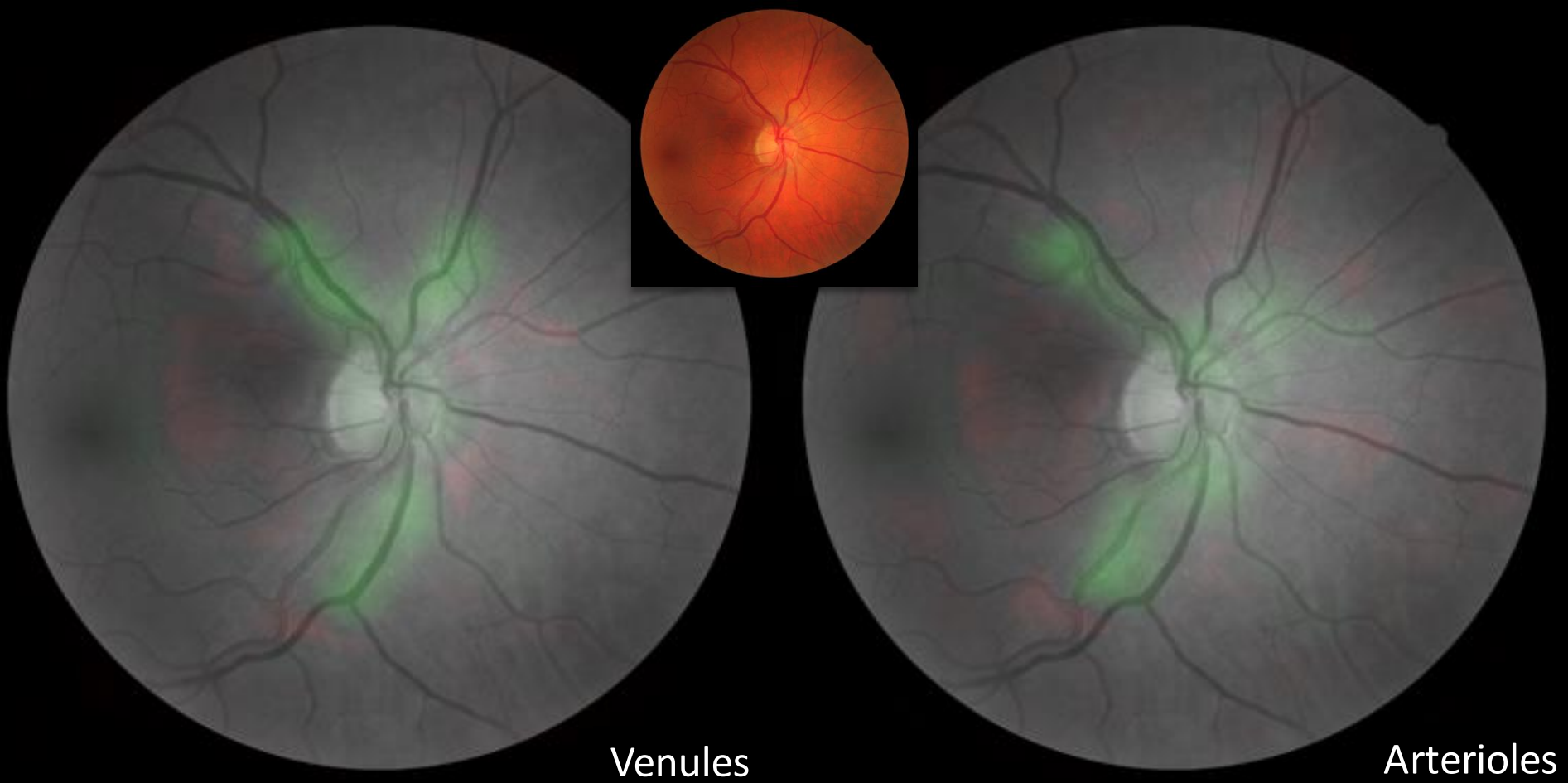
ResNet-50

- Replace final layer
 - # of output nodes = # of biomarkers
- Regression problem!
 - No activation function in the final layer



Results – Intra-Class Correlation





Example 3: Brain Tumor Segmentation

- MRI brain volumes (3D)
- FLAIR images
- Tumor segmentation

Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks

Hao Dong ^{†1}, Guang Yang ^{‡2,3}, Fangde Liu ^{‡1}, Yuanhan Mo ¹, Yike Guo ^{‡1}

¹ Data Science Institute, Imperial College London, SW7 2AZ, London, UK

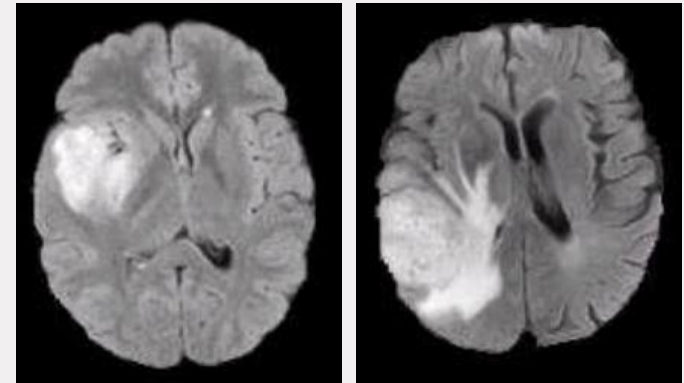
² Neurosciences Research Centre, Molecular and Clinical Sciences Institute,
St. George's, University of London, London SW17 0RE, UK

³ National Heart and Lung Institute, Imperial College London, SW7 2AZ, London, UK

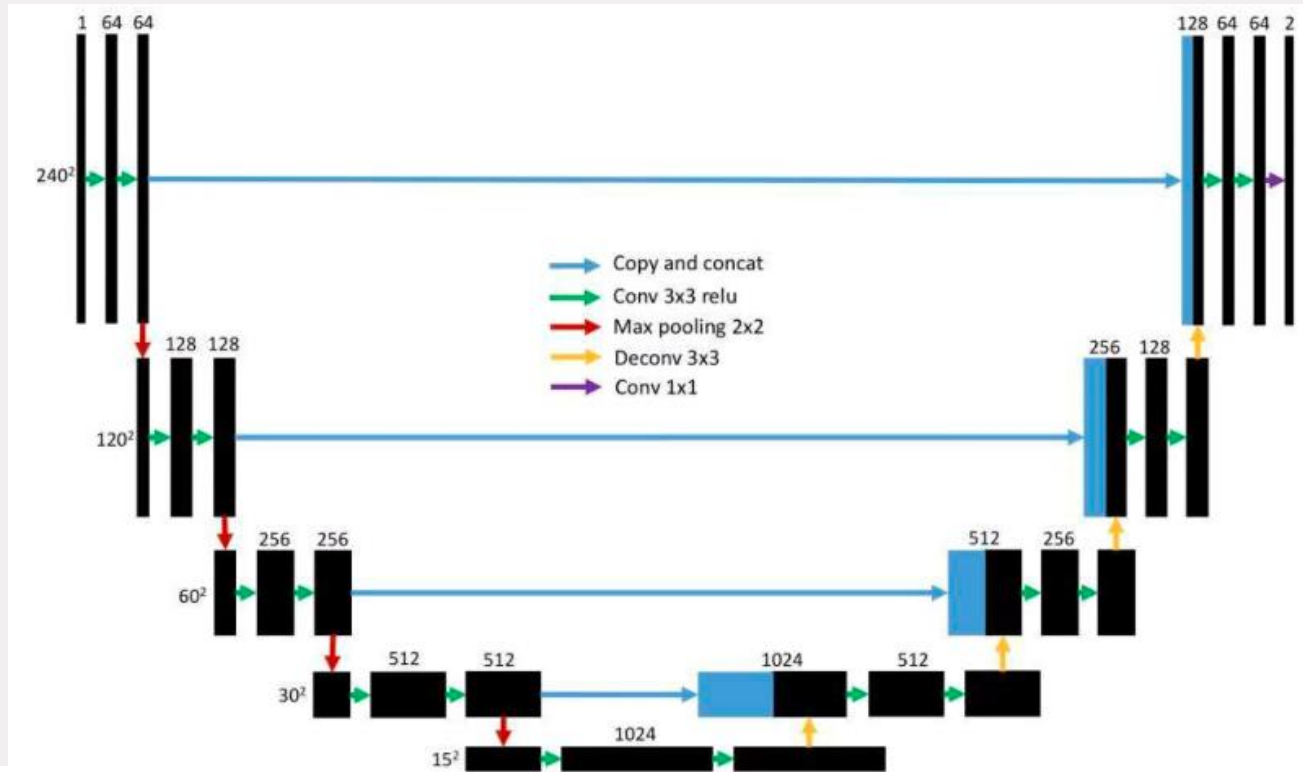
[†] Co-first authors contributed equally to this study.

[‡] Corresponding author.

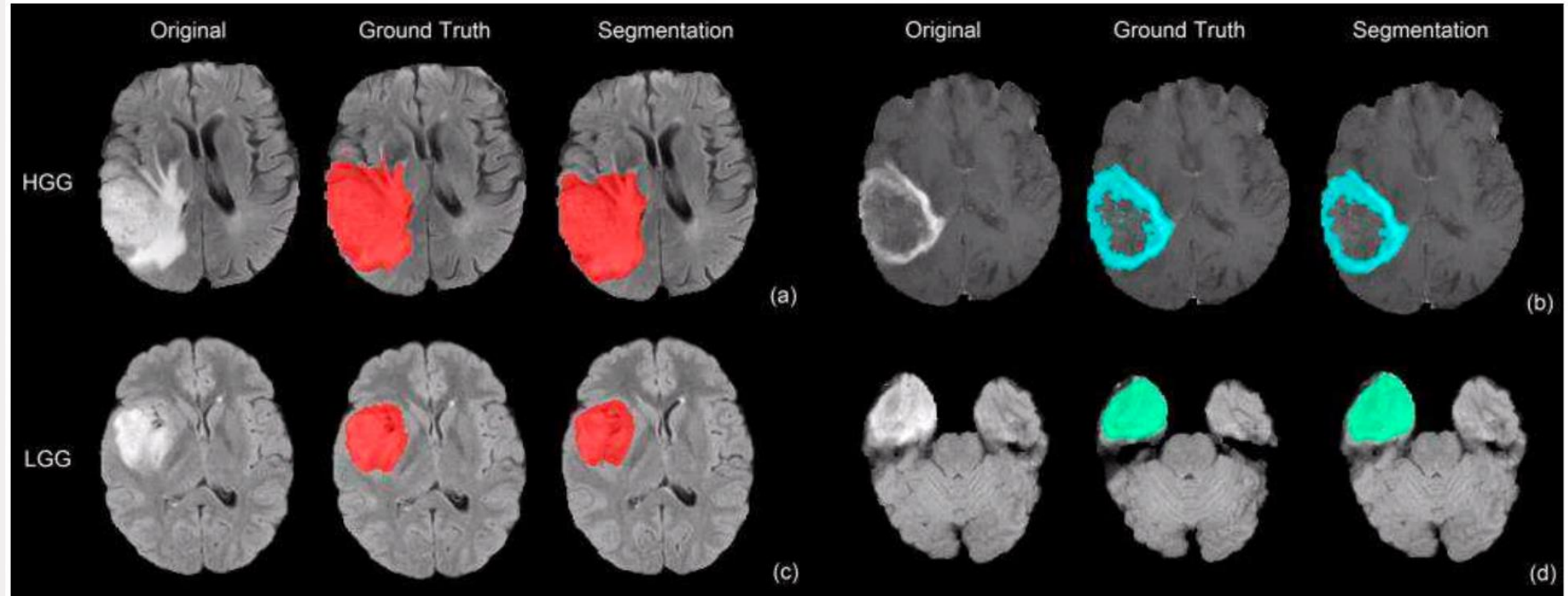
{hao.dong11, g.yang, fangde.liu, y.mo16, y.guo}@imperial.ac.uk



U-Net



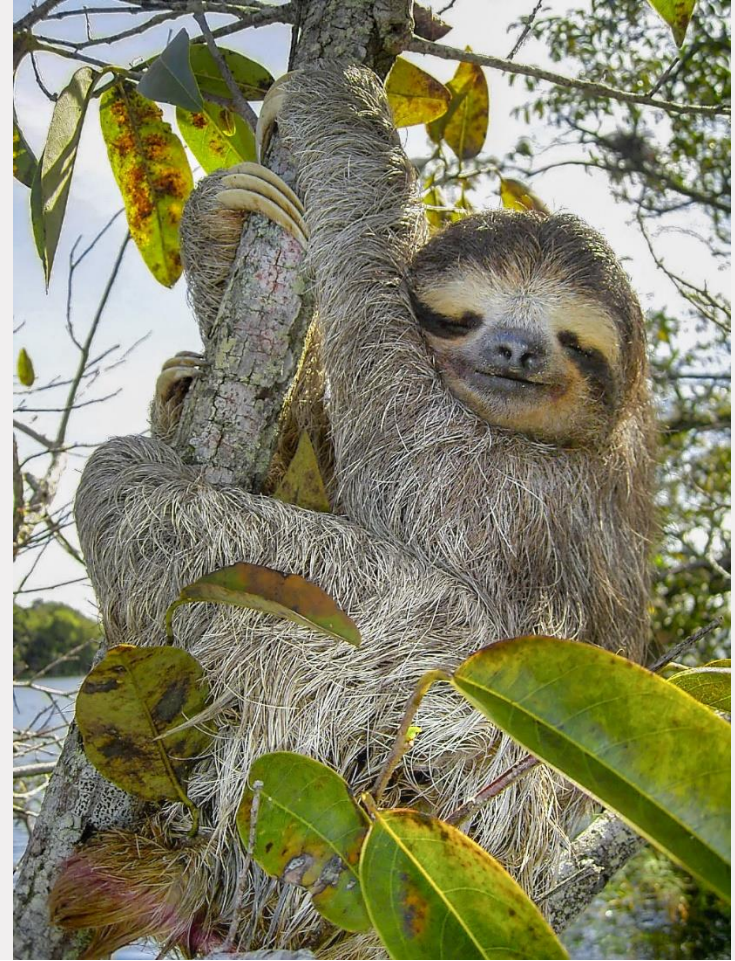
Segmentation results



Break!

Brown-throated sloth (Panama)

<https://en.wikipedia.org/wiki/Sloth>



Deep learning frameworks

- Python / Matlab / R
- Tensorflow / PyTorch / Theano [RIP] /
Deep learning toolbox for Matlab
- Keras / Sonnet
- Models on GitHub



Classify ImageNet classes with ResNet50

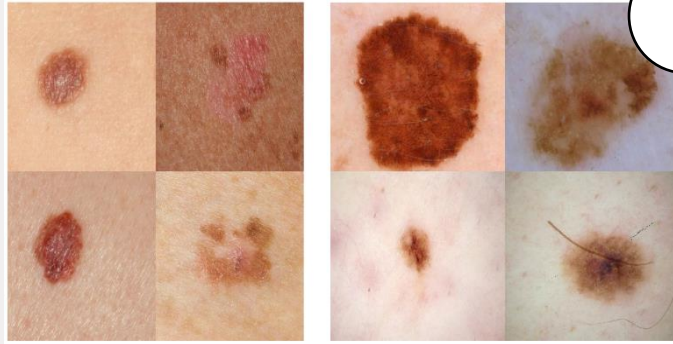
```
from keras.applications.resnet50 import ResNet50
from keras.preprocessing import image
from keras.applications.resnet50 import preprocess_input, decode_predictions
import numpy as np

model = ResNet50(weights='imagenet')

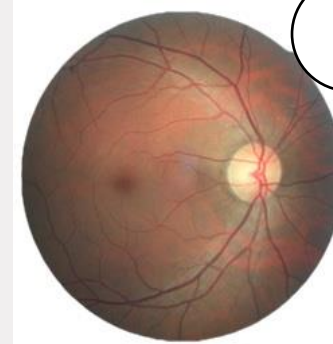
img_path = 'elephant.jpg'
img = image.load_img(img_path, target_size=(224, 224))
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

preds = model.predict(x)
# decode the results into a list of tuples (class, description, probability)
# (one such list for each sample in the batch)
print('Predicted:', decode_predictions(preds, top=3)[0])
# Predicted: [(u'n02504013', u'Indian_elephant', 0.82658225), (u'n01871265', u'tusker', 0.1122357),
              (u'n02504458', u'African_elephant', 0.061040461)]
```

Group exercise



1



2



3



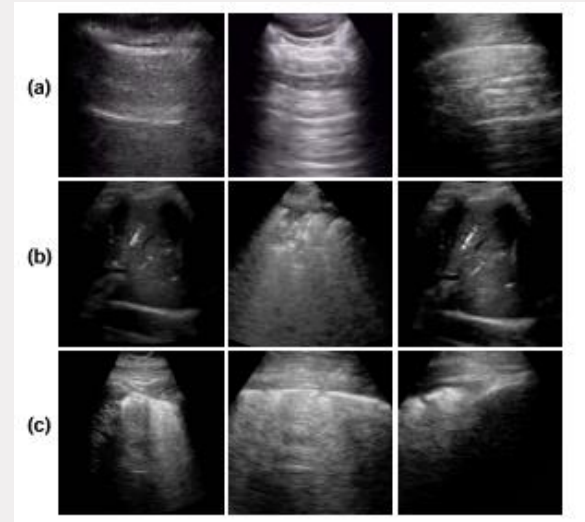
4

Group exercise

- Type of problem?
- Preprocessing steps?
- What kind of 'labels' do you need?
- What should the output look like?
- What should be the activation function in the final layer?
- What model do you choose?
- How do you evaluate performance?
- Other considerations?

Problem 0

COVID classification is a big problem at hand. You need to develop an algorithm for classification of COVID cases from healthy and pneumonia patients. How will you go about it ?



Awasthi et al., TUFFC 2021

Problem 0

- Classification (multi-class)
- Small app → small model and reduced input sizes for point of care use.
- Normalization of image intensities (many different transducers)
- Output model (in case of 3 classes): e.g. chance of belonging to normal class
- Activation function final layer CNN (in case of 3 classes): sigmoid
- Evaluate performance: e.g. classification accuracy

Problem 1

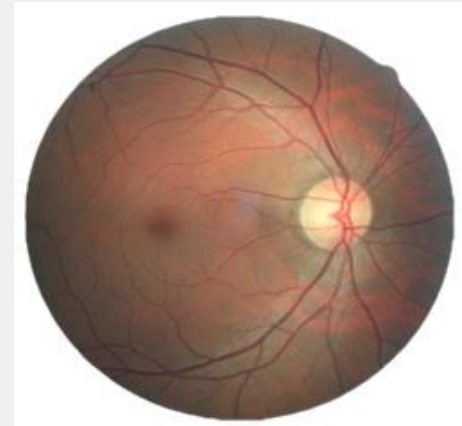
The Netherlands Cancer Institute (NKI) would like to develop an app that enables users to automatically assess skin lesions. The app should be deployable on any medium-end phone and all calculation should be done locally. The NKI has requested your group for general advice on the deep learning approach they should take.



Esteva et al., Nature 2017

Problem 2

Your team is competing in an online challenge. The objective: classification of retinal fundus images in one of five stages of diabetic retinopathy. Class 0 = no retinopathy. Classes 1-4 correspond to increasing severity of retinopathy. What is your approach?



<https://www.kaggle.com/c/diabetic-retinopathy-detection>

Problem 3

As part of a university project you are working on the development of an app that uses pictures of people's face to predict somebody's age. You can use the server of the university for the calculations.



<https://www.freecodecamp.org/news/how-to-build-an-age-and-gender-multi-task-predictor-with-deep-learning-in-tensorflow-20c28a1bd447/>

Problem 4

A vascular surgeon performs endovascular surgery using x-ray angiography. She would like to continuously see the veins and arteries clearly distinguished from the background on the monitor next to her. How would you approach this?



<https://blog.definitivehc.com>

Summary

- Different applications → different requirements
- History of deep learning models
- Deep learning enablers (computational power and data)
- Deep learning frameworks

Problem 1

The Netherlands Cancer Institute (NKI) would like to develop an app that enables users to automatically assess skin lesions. The app should be deployable on any medium-end phone and all calculation should be done locally. The NKI has requested your group for general advice on the deep learning approach they should take.



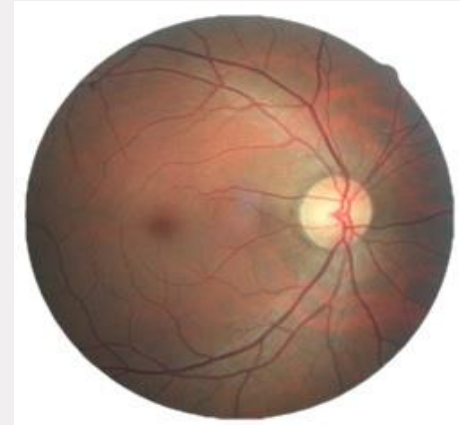
Esteva et al., Nature 2017

Problem 1

- Classification (2-class or multi-class)
- Small app → small model and reduced input sizes
- Normalization of image intensities (many different camera's)
- Output model (in case of 2 classes): e.g. chance of belonging to malignant class
- Activation function final layer CNN (in case of 2 classes): sigmoid
- Evaluate performance: e.g. classification accuracy

Problem 2

Your team is competing in an online challenge. The objective: classification of retinal fundus images in one of five stages of diabetic retinopathy. Class 0 = no retinopathy. Classes 1-4 correspond to increasing severity of retinopathy. What is your approach?



<https://www.kaggle.com/c/diabetic-retinopathy-detection>

Problem 2

- Classification (multi-class)
- Normalization of image intensities (many different camera's)
- Output model: e.g. chance of belonging to each class, and/or class with highest probability
- Activation function final layer CNN: softmax (not needed for exam)
- Evaluate performance: e.g. classification accuracy. One could take into account the extent of the error. Predicting class 0 while label is class 4 results in a bigger error than predicting class 3.
- To achieve high classification accuracy one could consider an ensemble of state-of-the art deep learning models

Problem 3

As part of a university project you are working on the development of an app that uses pictures of people's face to predict somebody's age. You can use the server of the university for the calculations.



<https://www.freecodecamp.org/news/how-to-build-an-age-and-gender-multi-task-predictor-with-deep-learning-in-tensorflow-20c28a1bd447/>

$$\hat{y} = \theta^T \mathbf{x}$$

Problem 3

- Regression, as the output is continuous. However, classification (of binned age-groups) is also possible
- As preprocessing, one could crop the image to only the person's face. For this you could use user-input.
- Output model: predicted age, perhaps with boundaries (any number between 0 and 100)
- Activation function final layer CNN: none, since
- Evaluate performance: Euclidean distance between model predictions and actual age

Problem 4

A vascular surgeon performs endovascular surgery using x-ray angiography. She would like to continuously see the veins and arteries clearly distinguished from the background on the monitor next to her. How would you approach this?



<https://blog.definitivehc.com>

Problem 4

- Segmentation problem
- As preprocessing, one could reduce the number of frames
- Output model: image (mask) that shows locations of veins and vessels. It is possible to output multiple maps (one with background, one with veins and one with arteries).
- It could be approached as a pixel-wise classification problem. If so, the activation function of the final layer could be a sigmoid for veins vs background. Or a softmax for multiple classes.
- Evaluate performance: e.g. Dice
- Type of model: U-Net would be good starting point