딥러닝을 이용한 초음파 영상 진단 모델 개발

Eunjung Lee

School of Mathematics and Computing

(Computational Science and Engineering)

Yonsei University

Computational Science & Engineering

연세대학교 YONSEI UNIVERSITY

Outline

- SERA 소개
- Medical Imaging
- CADx
- Classical approaches + Machine learning
- Deep learning algorithms used for feature extraction and classification
- SERA 상세소개

SERA

– Severance Diagnostic Helper based on Deeplearning designed by Yonsei-CSE with Matlab

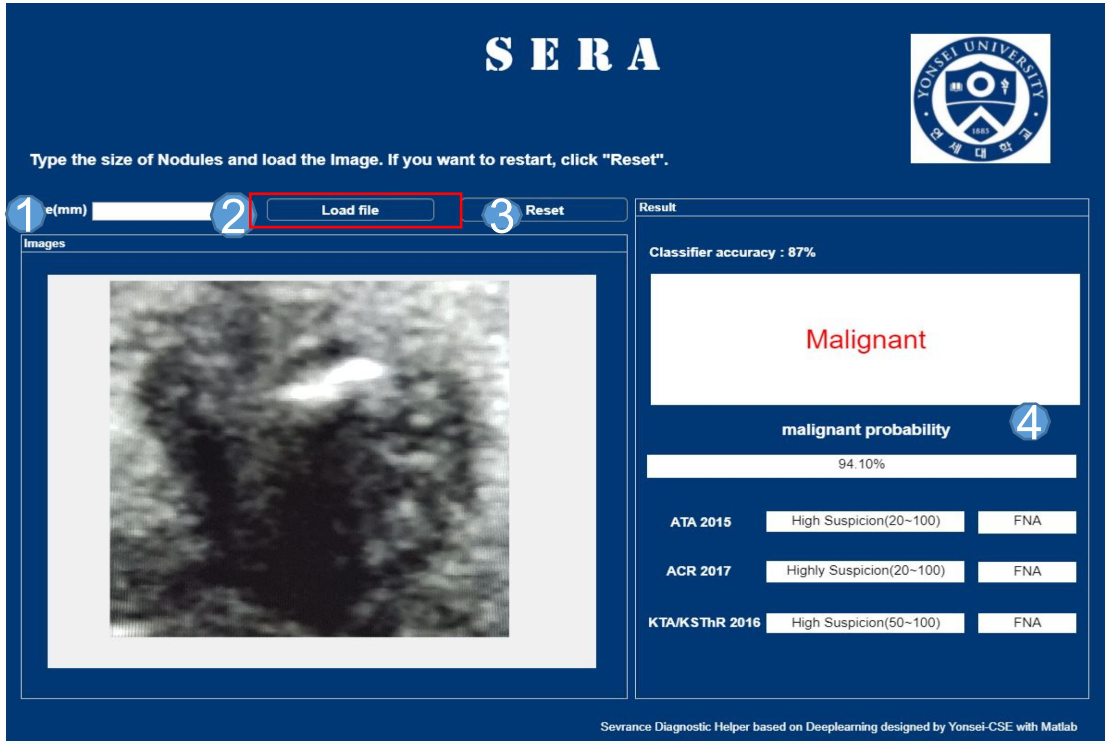


fig1)computer application

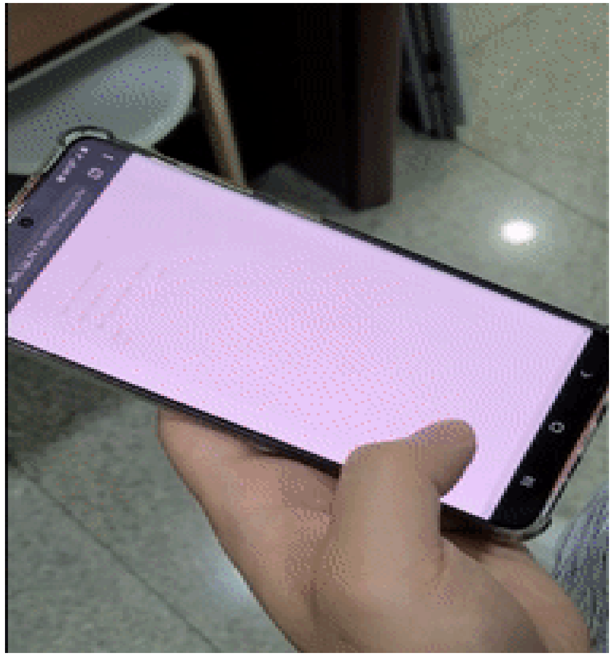


fig2) mobile application

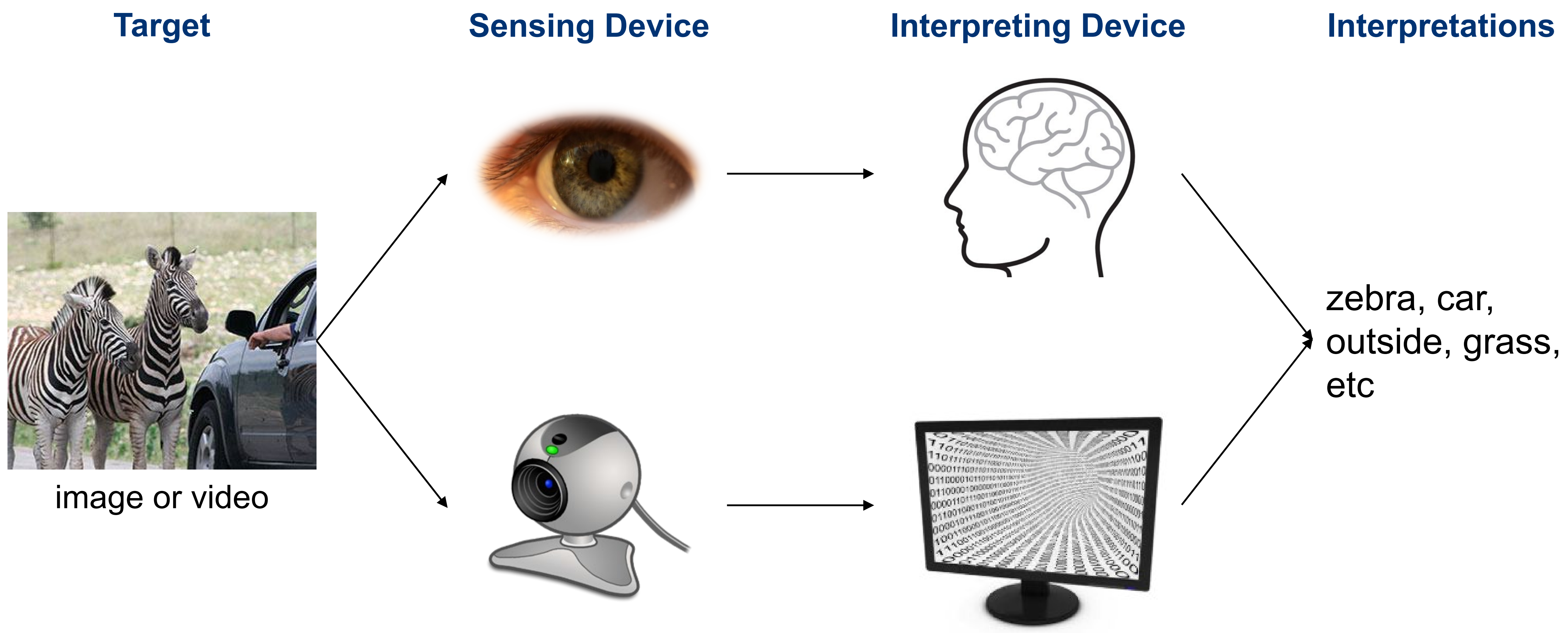
1 In put the node size

2 click and upload the image

3 If you want to reset, click

4 you can see the results of the tumor and whether you need to FNA

Human Vision and Computer Vision



Some Computer Vision Tasks

- Exam classification
  - Given only one label and assume there is only one object.
  - Determine whether or not given image is the label.
- Object classification / localization
  - Given multiple labels and assume there is only one object.
  - Determine the optimal label for given image (classification) and find the object's location (localization).
- Object detection: object localization for multiple objects
  - Given multiple labels.
  - Determine the optimal label for each object in given image and find each object's location.

Exam Classification

Is there a cat?

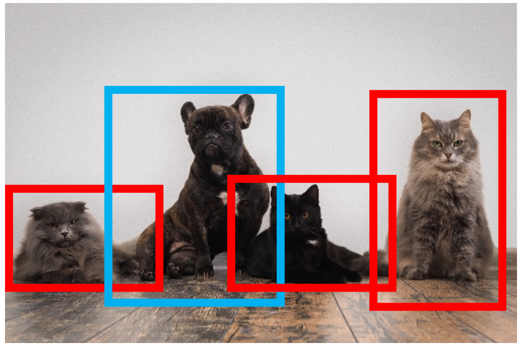
Object Classification

Is it cat, dog, or lion?

Object Localization

Is it cat, dog, or lion?
Where is it?

Object Detection



Are they cat, dog, or lion?
Where are they?

Some Computer Vision Tasks

- Exam classification

  Given only one label and assume there is only one object.

  Determine whether or not given image is the label.

- Object classification / localization

  Given multiple labels and assume there is only one object.

  Determine the optimal label for given image (classification) and find the object's location (localization).

- Object detection: object localization for multiple objects

  Given multiple labels.

  Determine the optimal label for each object in given image and find each object's location.

Exam Classification

Is there a cat?

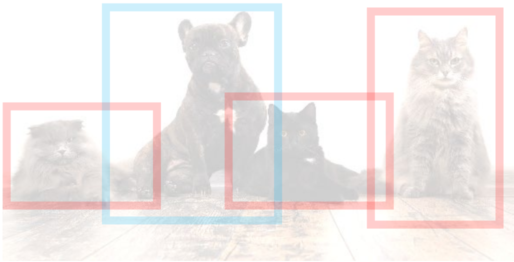
Object Classification

Is it cat, dog, or lion?

Object Localization

Is it cat, dog, or lion?
Where is it?

Object Detection



Are they cat, dog, or lion?
Where are they?

We focus on the object classification problem.

Object Classification in Medical Imaging

In medical imaging, classification problem is said as computer-aided diagnosis (CADx).

- Medical images

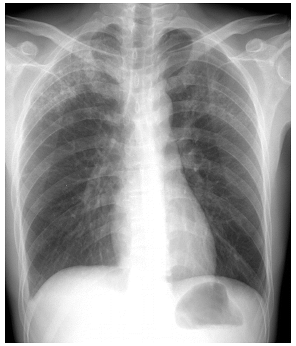
  X-ray, CT, MRI or ultrasound

- Labels

  Most problems are binary: (benign, malignant) or (normal, abnormal)

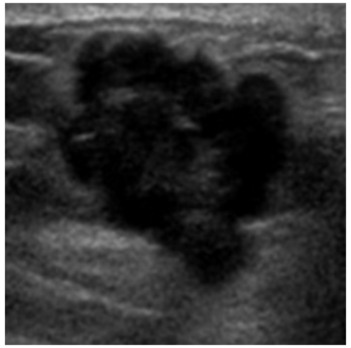
Pulmonary tuberculosis(X-ray) healthy or not?

Lakhani et al, Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks (2017)



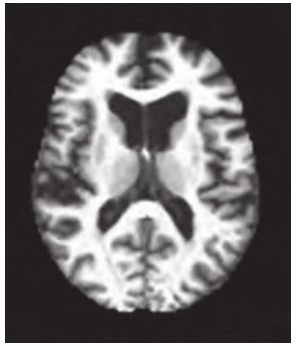
Lung nodule(CT) benign or malignant?

Kumar et al, Lung nodule classification using deep features in CT images (2015)



Alzheimer's disease(MRI) normal or Alzheimer?

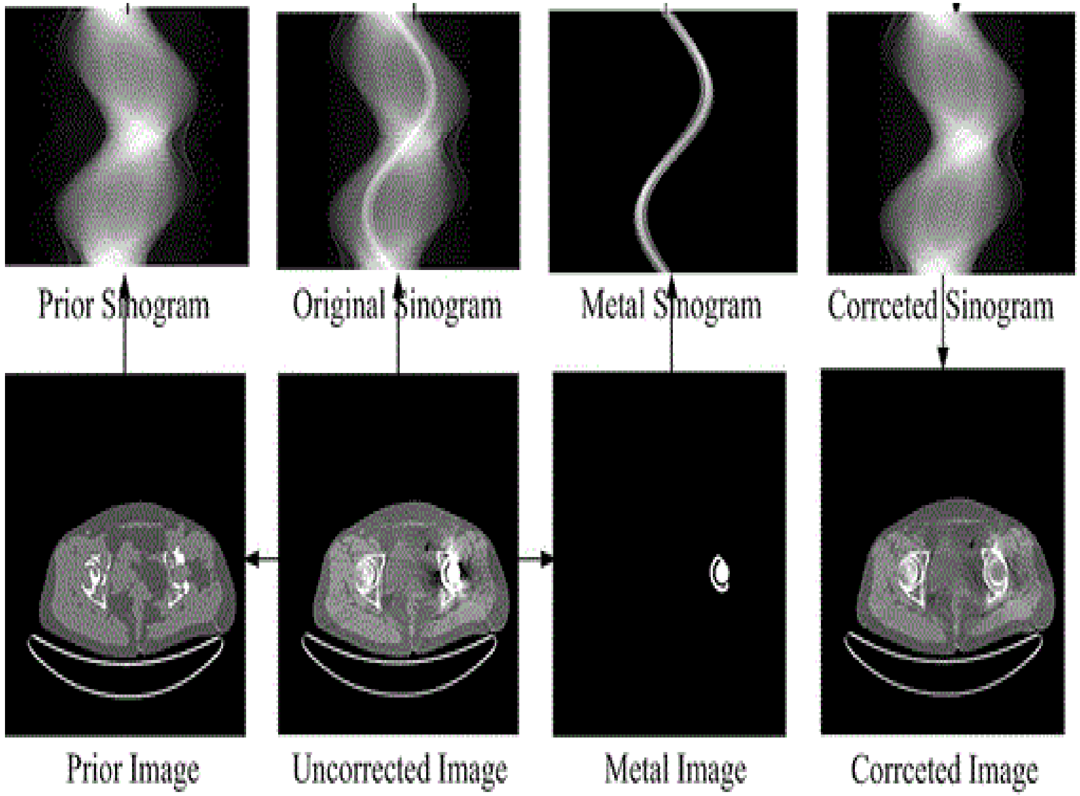
Jha et al, Alzheimer’s disease detection using extreme learning machine, complex dual tree wavelet principal coefficients and linear discriminant analysis (2018)



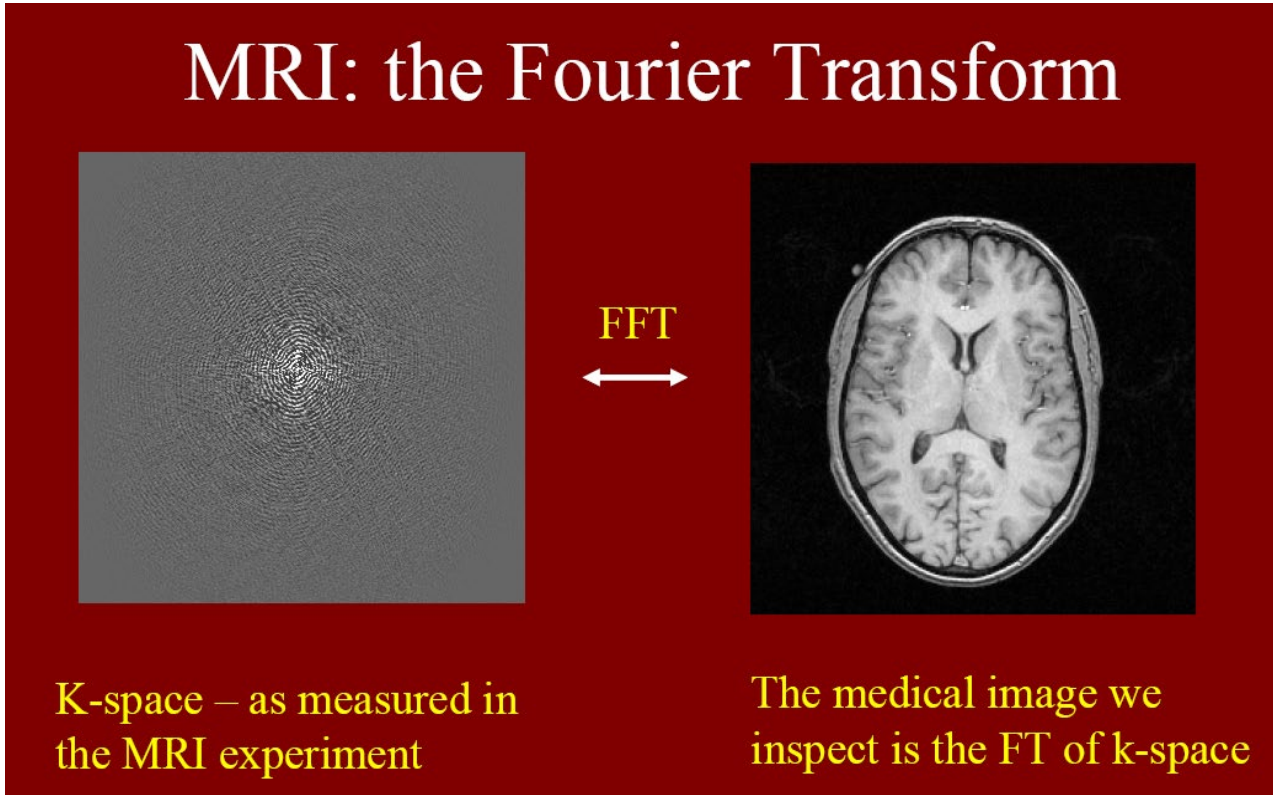
Breast tumor(ultrasound) benign or malignant?

Han et al, A deep learning framework for supporting the classification of breast lesions in ultrasound images (2017)

Preprocessing –data collection to generate the image

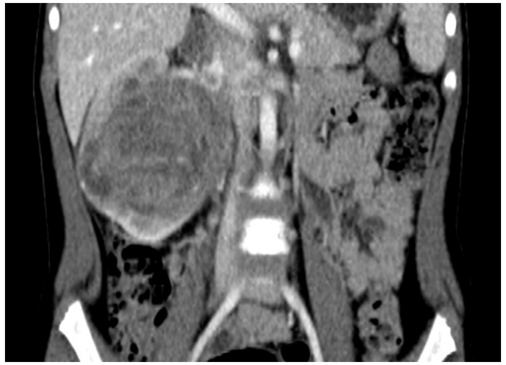


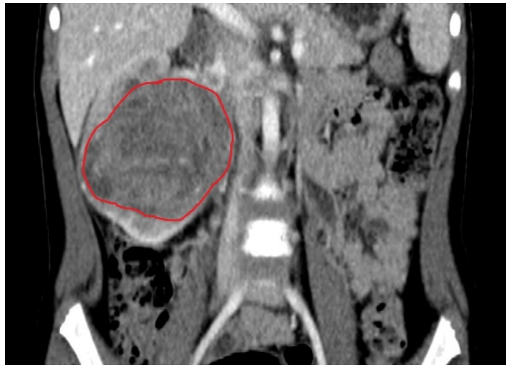
Gaussian diffusion sinogram inpainting for X-ray CT metal artifact reduction, Chengtao Peng, Bensheng Qiu, Ming Li, Yihui Guan, Cheng Zhang, Zhongyi Wu and Jian Zheng, BioMedical Engineering OnLine (20171) 6:1



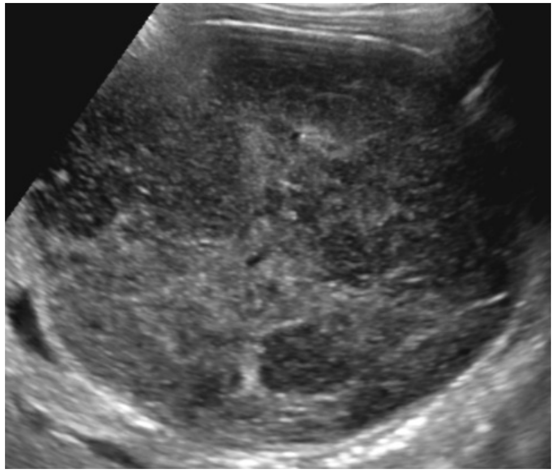
$$S(t) = \iint \rho(x,y)\ \exp\{2\pi i(k_x x + k_y y)\}\ dxdy$$

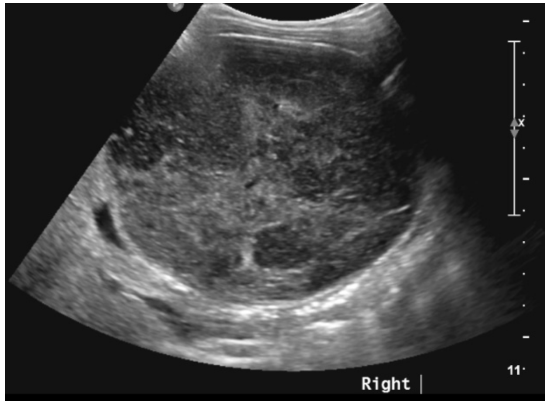
Preprocessing - ROI extraction

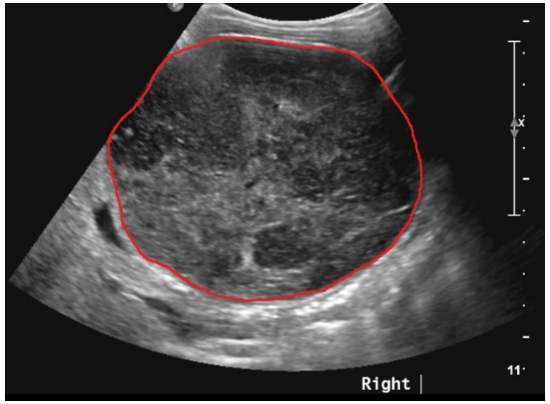




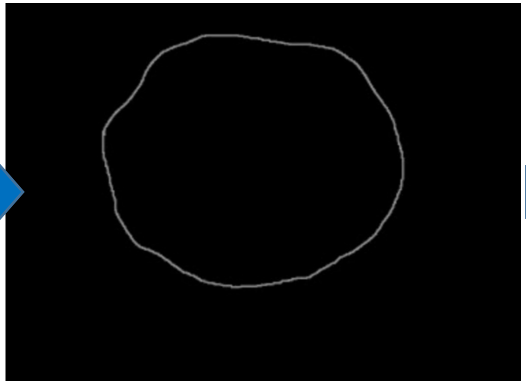
Before normalization

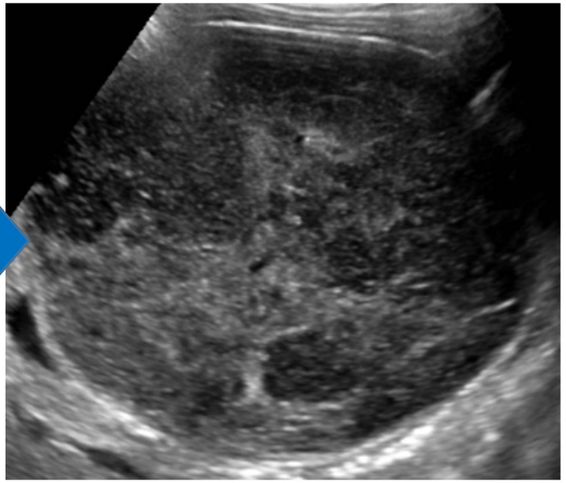


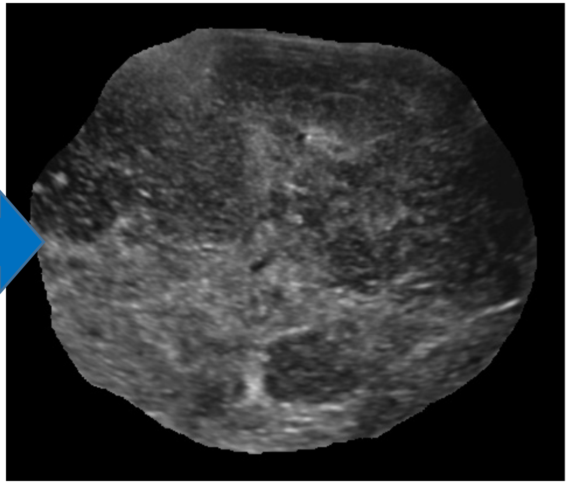




ROI marker







Normalization

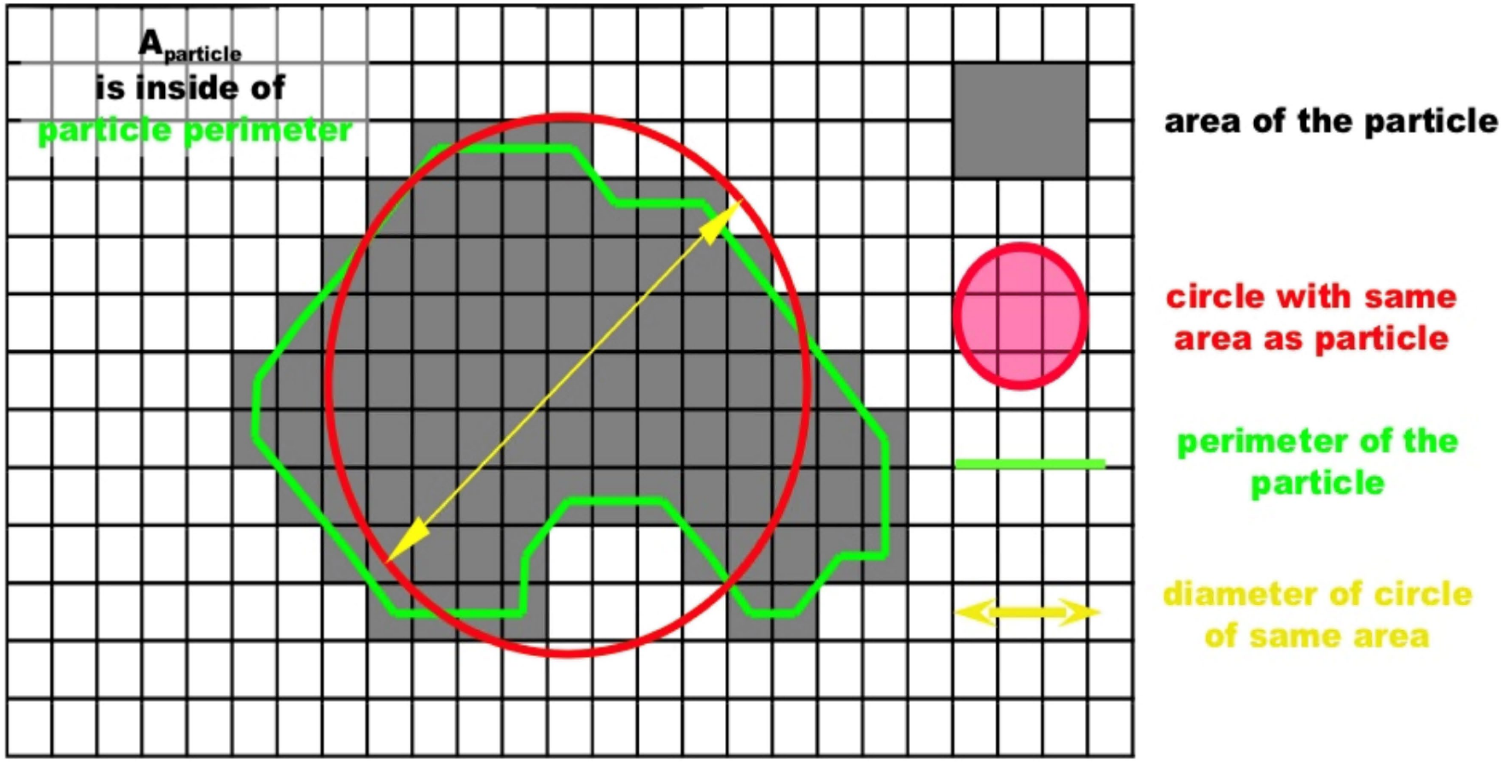
Feature Extraction

Feature extraction involves extracting information and generating features from original data.

Feature extraction

- **Shape based features**
  - Compactness
  - Area
  - Roundness
  - Perimeter
  - Circularity
  - Sphericity
  - Convexity
  - …
- **Texture based features**
  - 1st order statistics
    - Energy, Entropy
    - Mean, Variance
    - Standard deviation
    - Skewness, Median
    - …
  - 2nd order statistics
    - GLCM
    - GLRLM
    - …
  - **Wavelets**

Example of Shape Based Features: Particle



$A_{green} \sim 77\,\text{pixel},\ \text{diameter}_{red} = 10\,\text{pixel},\ \text{perimeter}_{green} \sim 38{,}5\,\text{pixel},$

$\textbf{circularity} \sim 31/38{,}5 = \mathbf{0.81},\ \textbf{sphericity} = 4 * \pi * 77/(38.5)^2 = \mathbf{0.81}^2 = \mathbf{0.65}$

Example circularity, sphericity of a particle

circularity = perimeter of circle / perimeter of particle

sphericity = $4\pi \times$ area of particle / (perimeter of particle)^2 = (circularity)^2

Bodycomb, Image analysis: Evaluating particle shape (HORIBA webinar, 2011)

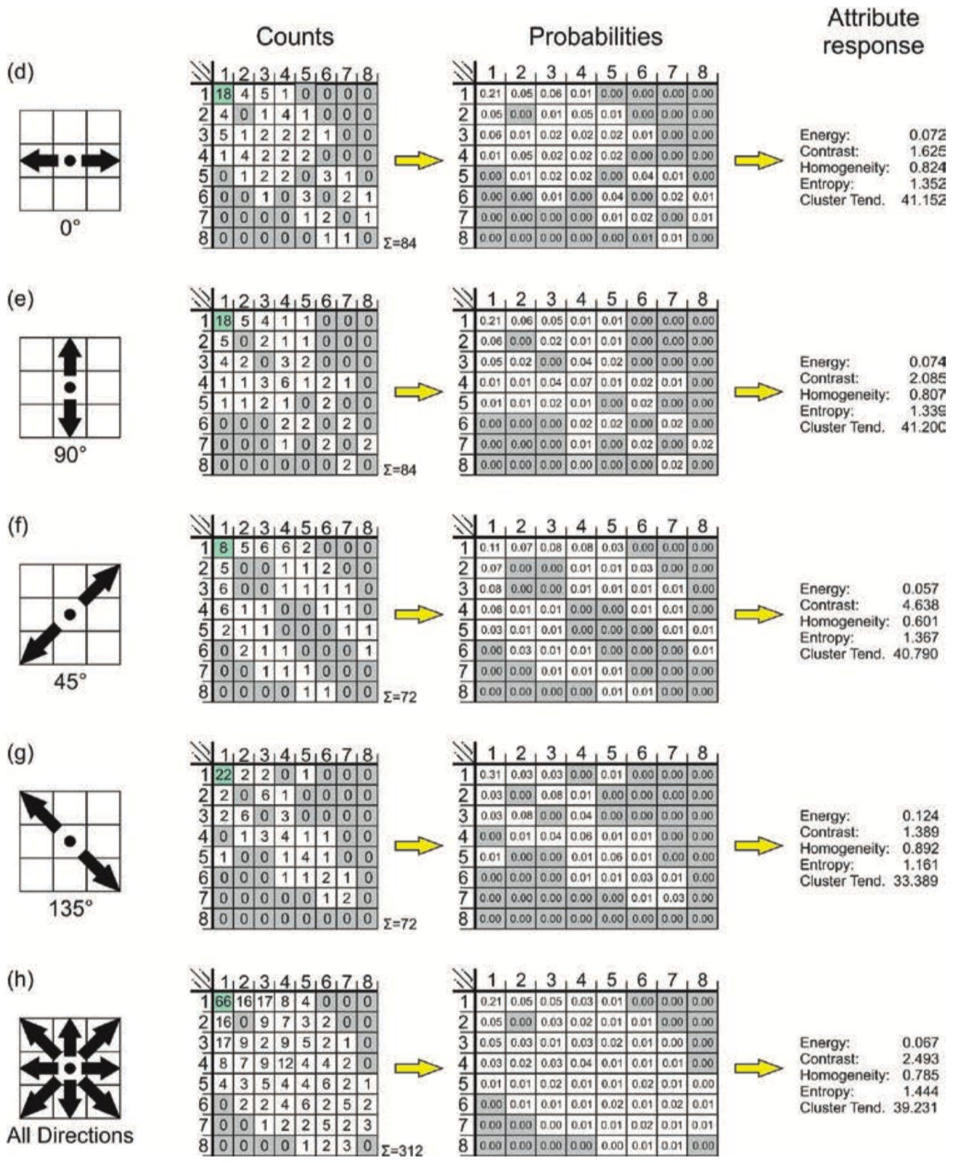
Example of Texture Based Features: GLCM

- Grey Level Co-occurrence Matrix (GLCM)

  matrix describing 2nd order joint probability function

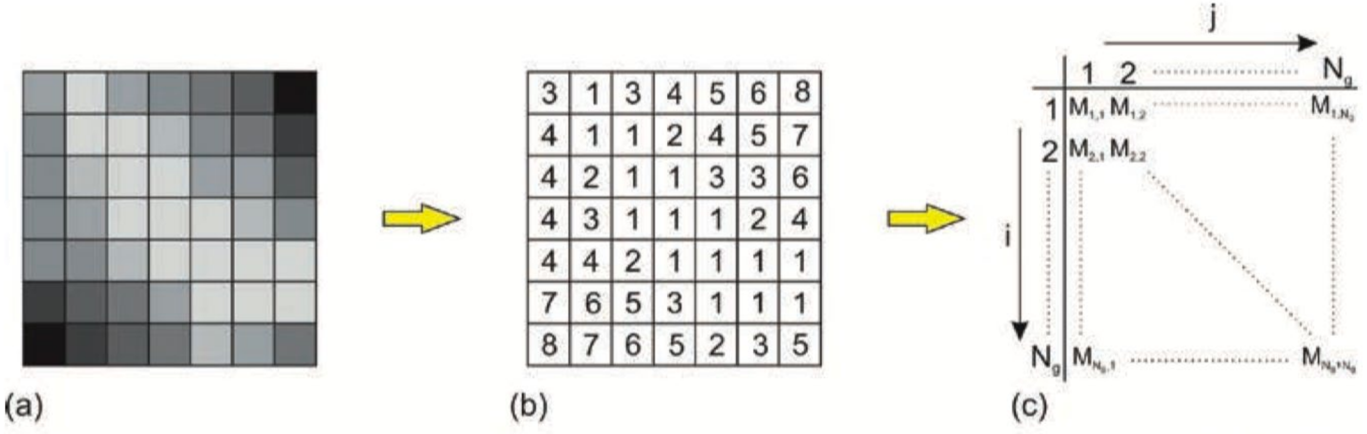
  If the image has 16 grey levels, then convert it as 4 bits.

  If the image has 256 grey levels, then convert it as 8 bits.



Example Converted image (a) and its discrete value (b). GLCM is produced by index (c). GLCM is determined for horizontal (d), vertical (e), 45° diagonal (f), 135° diagonal (g), and all directions (h).

Eichkitz et al, Grey level co-occurrence matrix and its application to seismic data (2015)



Example of Texture Based Features: Wavelets

Shear-wave elastography

: breast masses (GangNam Severance)

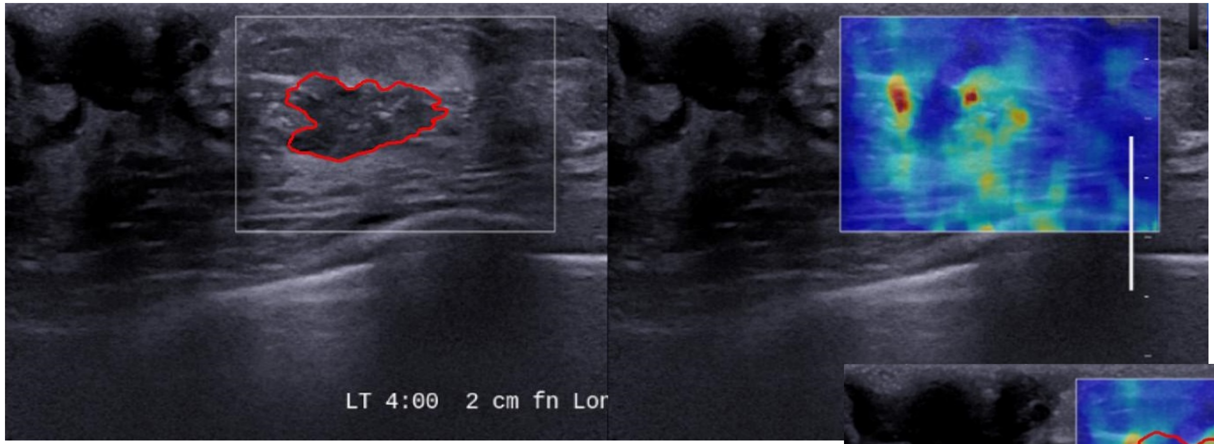
A single level discrete 2D wavelet transformation

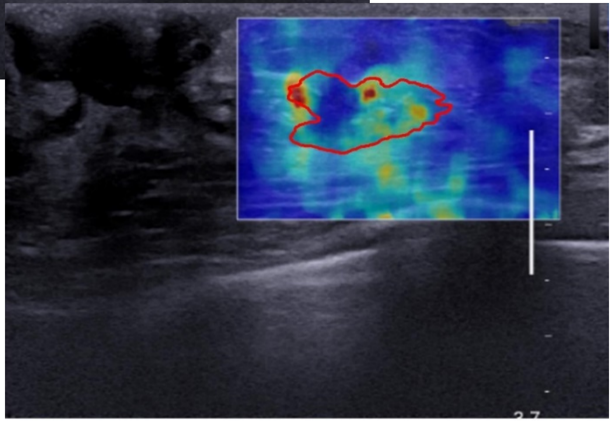
$$X_{LH}(i,j) = \sum_{p}\sum_{r} L(p)H(r)X(i+p, j+r)$$

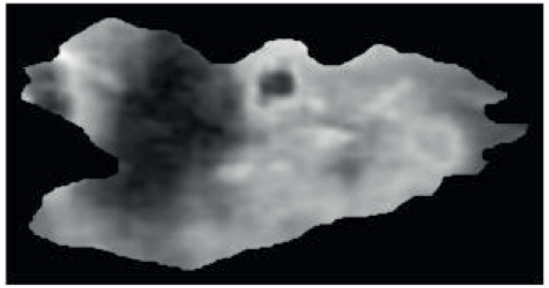
: low/low pass (LL), low/high pass (LH),

high/low pass (HL), high/high pass (HH) filters

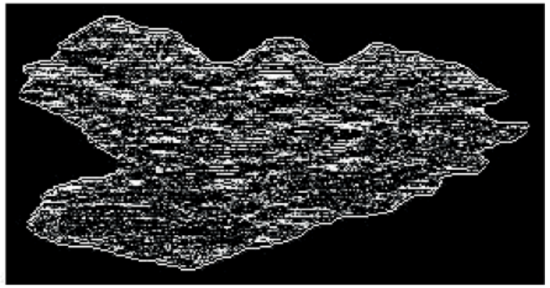
SWE images are composed of gray-scale (left) and color-coded elasticity images (right)





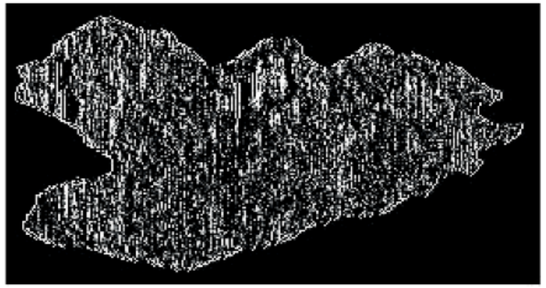


LL

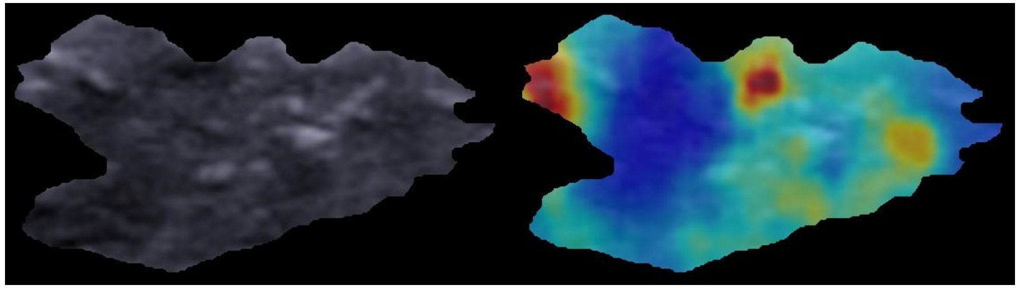


LH

HL

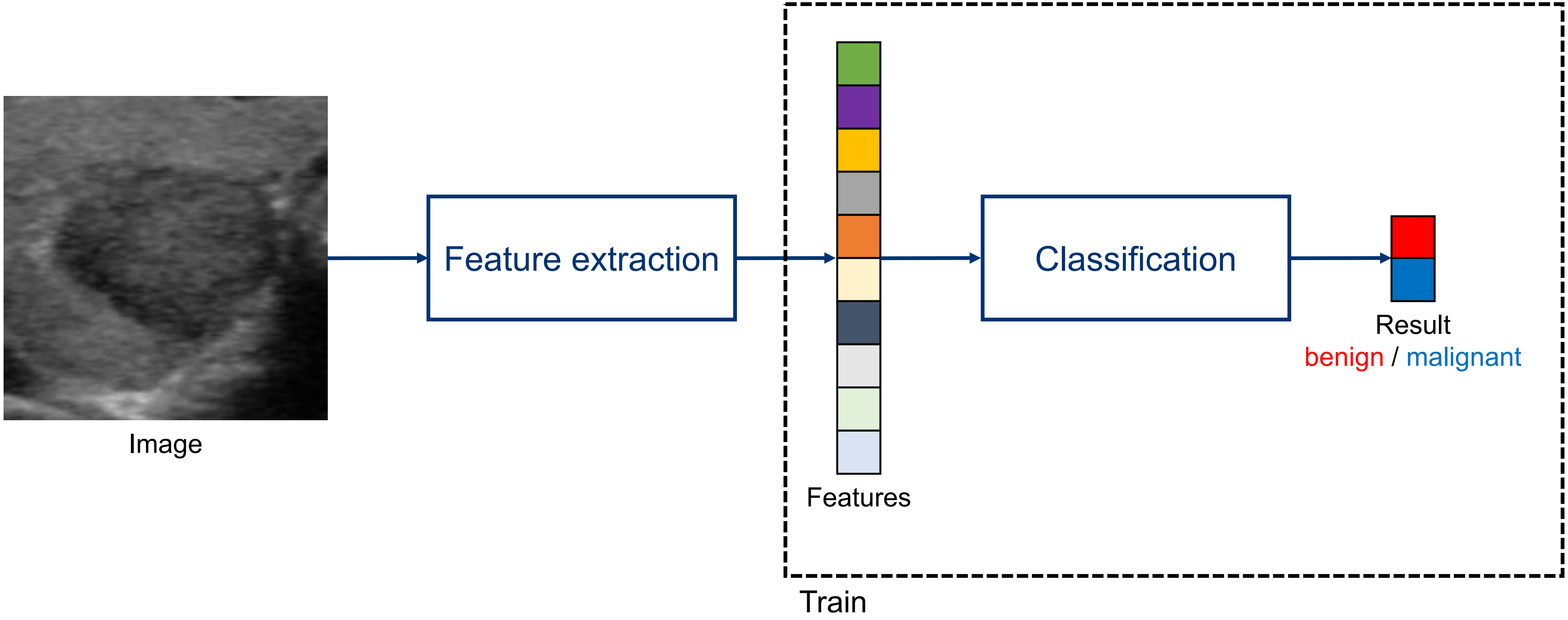


HH



Classification

Classification integrates extracted features and then estimates the class of data.



Example of Classification: Support Vector Machine

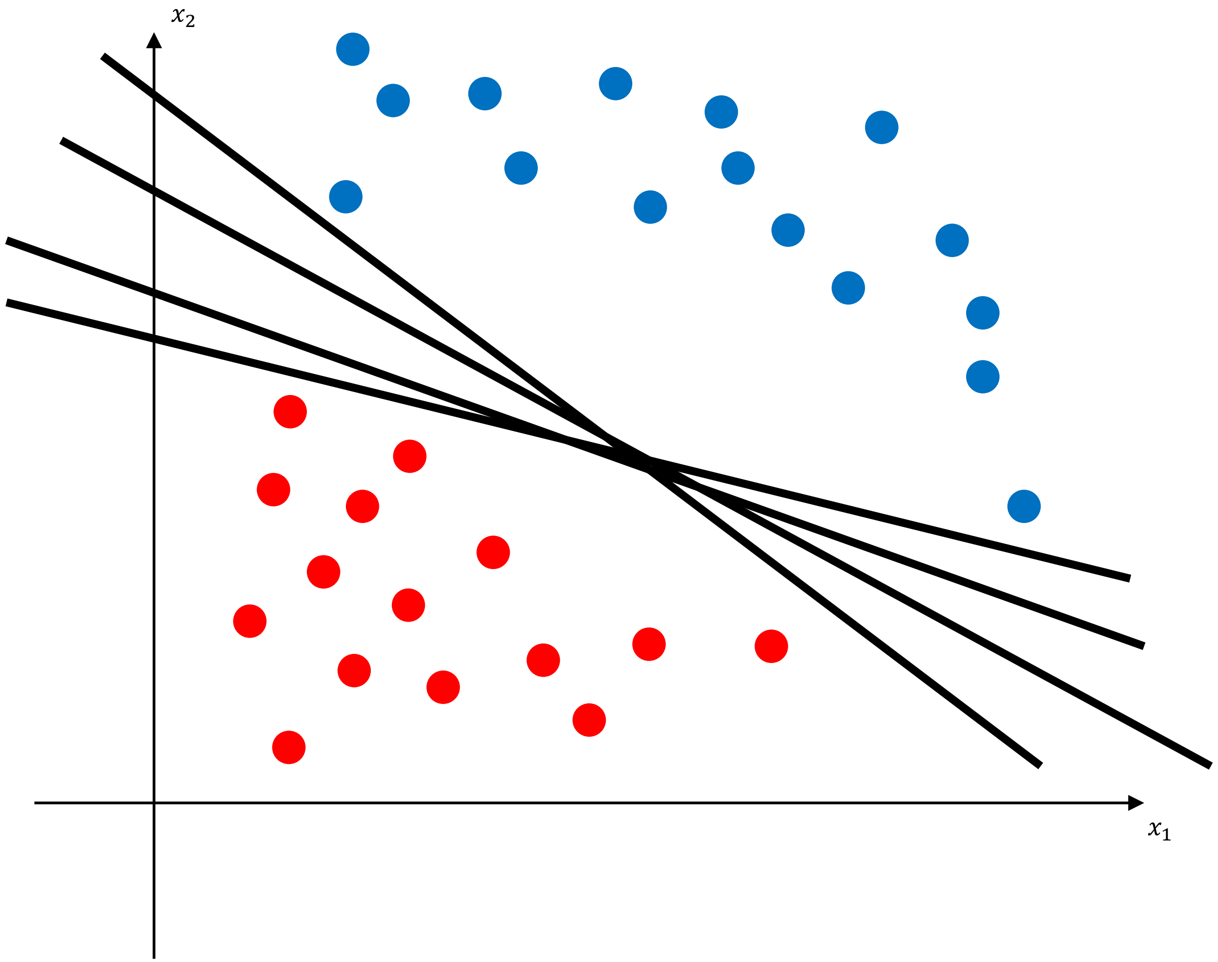
- Assume

  features are in two dimension.

  red/blue circles are labeled as benign/malignant.

- How to separate them?

  There are many lines to separate them perfectly.



Example of Classification: Support Vector Machine

- Assume

  features are in two dimension.

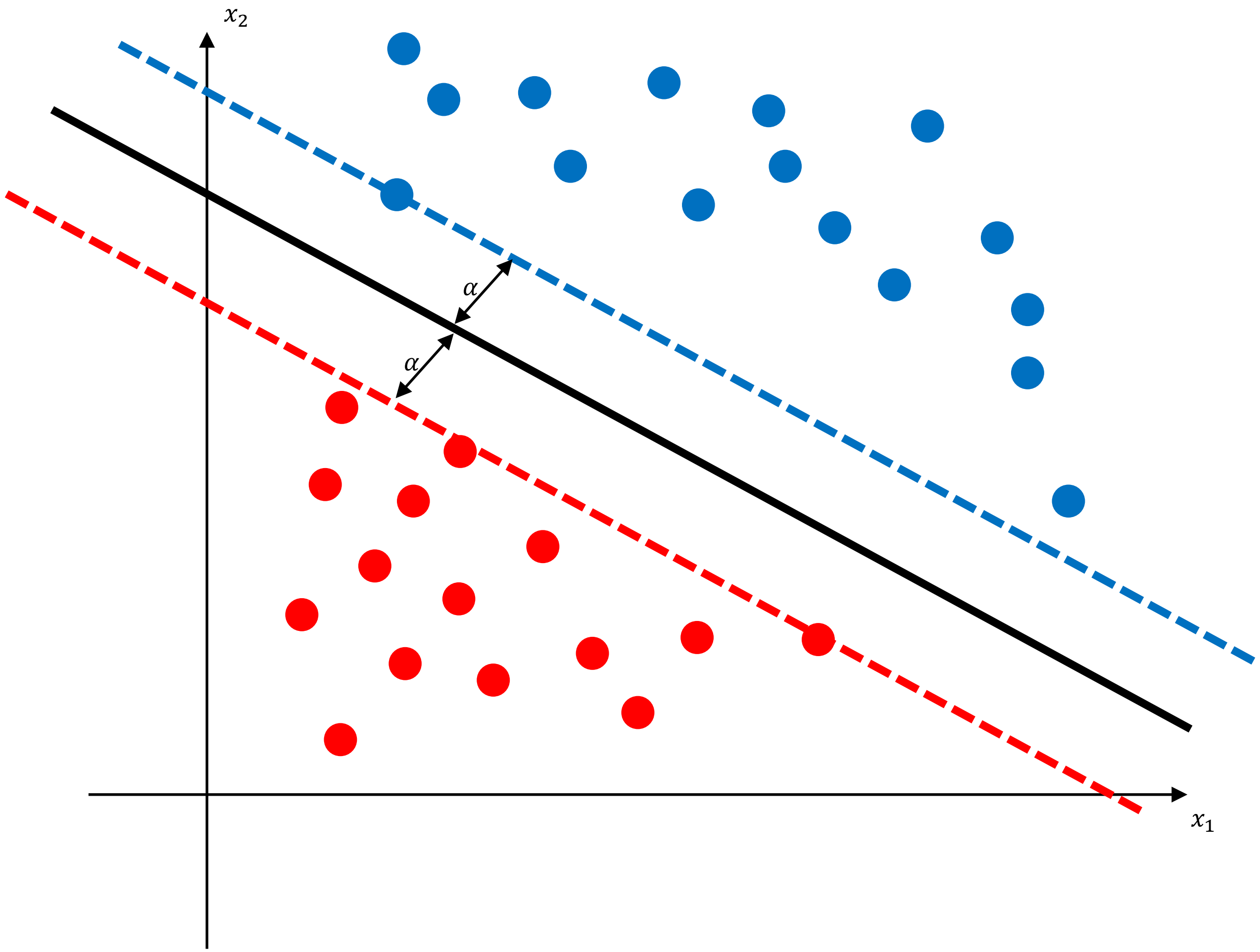
  red/blue circles are labeled as benign/malignant.

- How to separate them?

  There are many lines to separate them perfectly.

- Consider the margin $\beta = 2\alpha$.

  Find the line which has the largest β.



Example of Classification: Support Vector Machine

- Assume

  features are in two dimension.

  red/blue circles are labeled as benign/malignant.

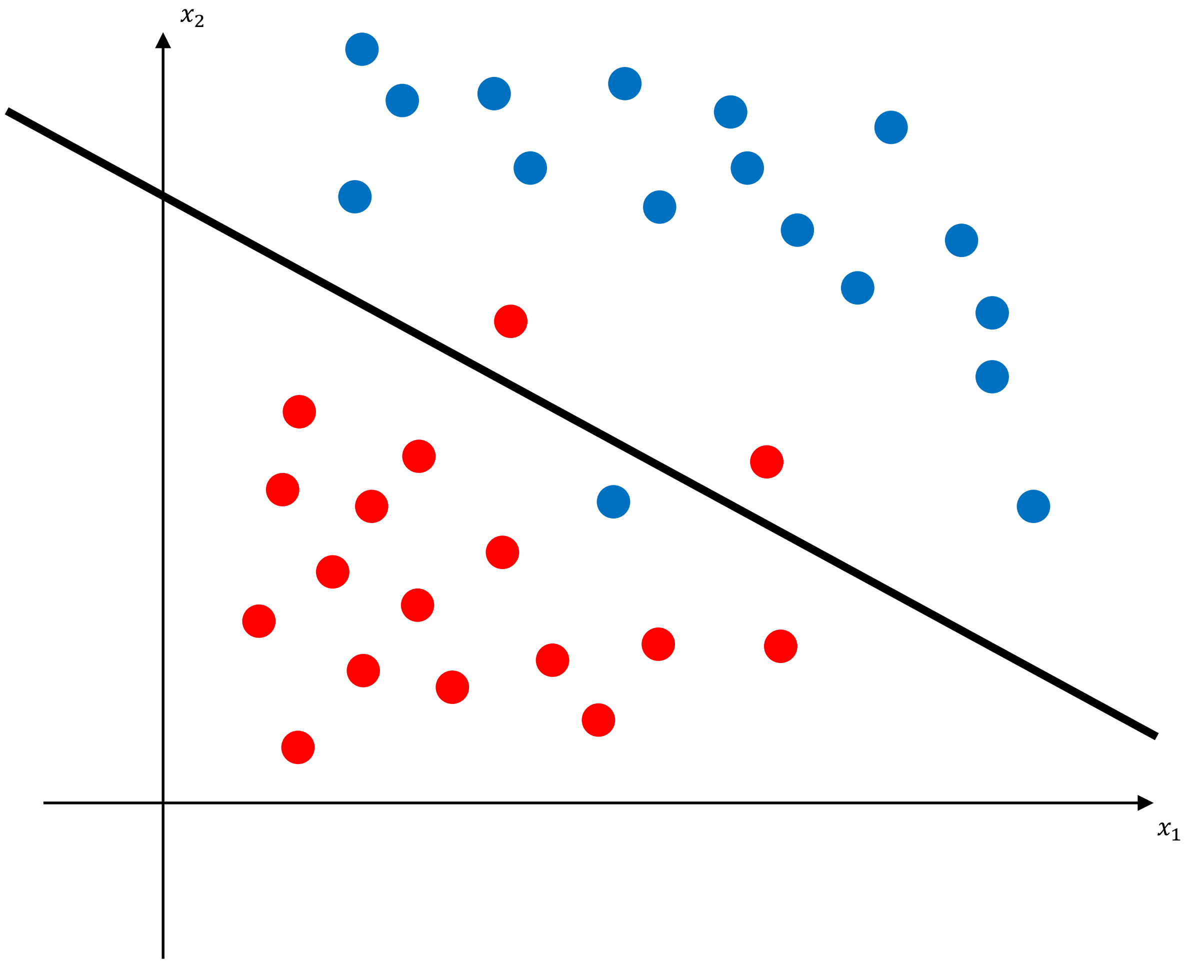
- How to separate them?

  There are many lines to separate them perfectly.

- Consider the margin $\beta = 2\alpha$.

  Find the line which has the largest β.

- Sometimes there is no line which separate them perfectly.



Example of Classification: Support Vector Machine

- Assume

  features are in two dimension.

  red/blue circles are labeled as benign/malignant.

- How to separate them?

  There are many lines to separate them perfectly.

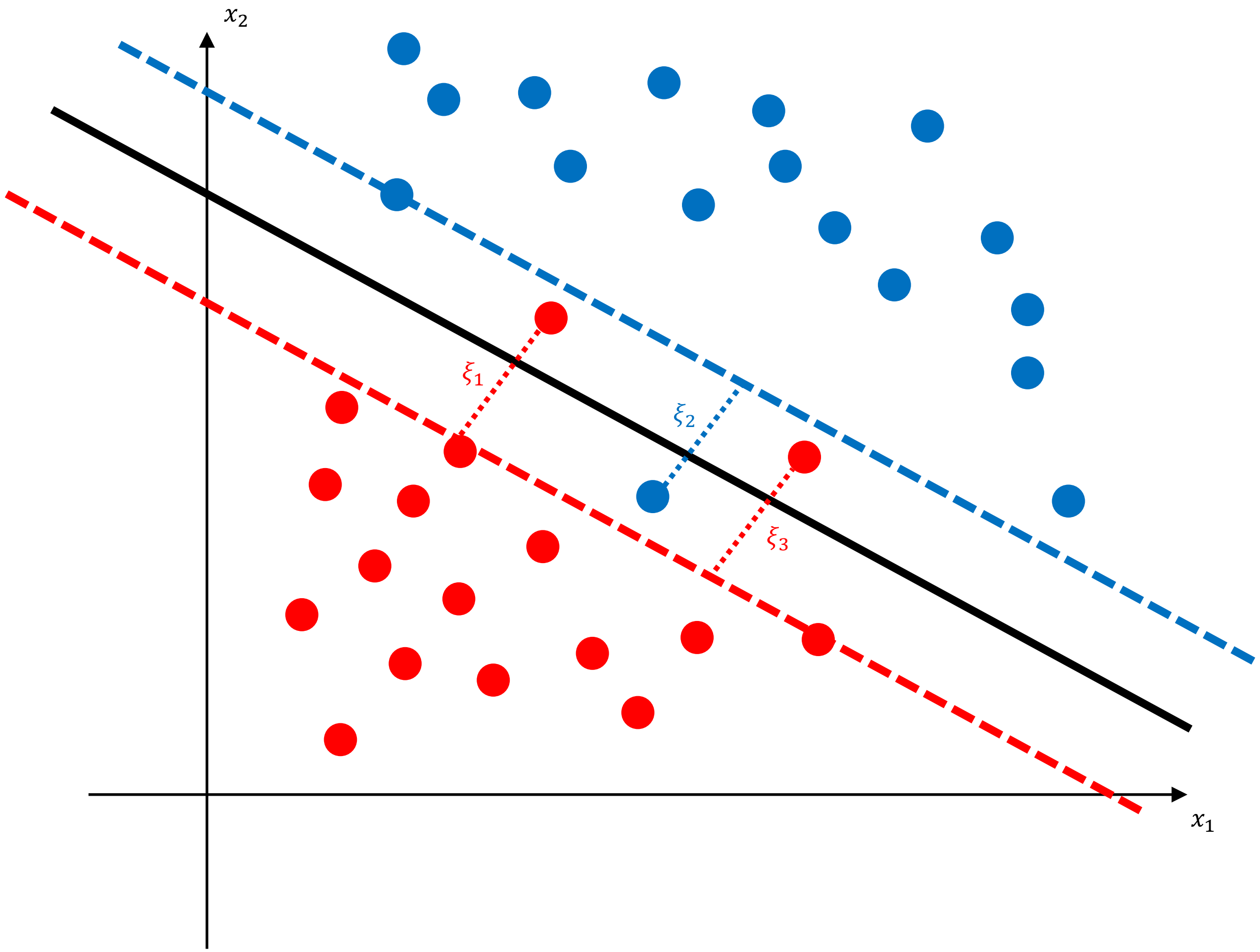
- Consider the margin $\beta = 2\alpha$.

  Find the line which has the largest β.

- Sometimes there is no line which separate them perfectly.

  One method(soft-margin) is to allow errors.

  We need to choose the tradeoff parameter $C > 0$ related to ξ_i.



Example of Classification: Support Vector Machine

- Assume

  features are in two dimension.

  red/blue circles are labeled as benign/malignant.

- How to separate them?

  There are many lines to separate them perfectly.

- Consider the margin $\beta = 2\alpha$.

  Find the line which has the largest β.

- Sometimes there is no line which separate them perfectly.

  One method(soft-margin) is to allow errors.

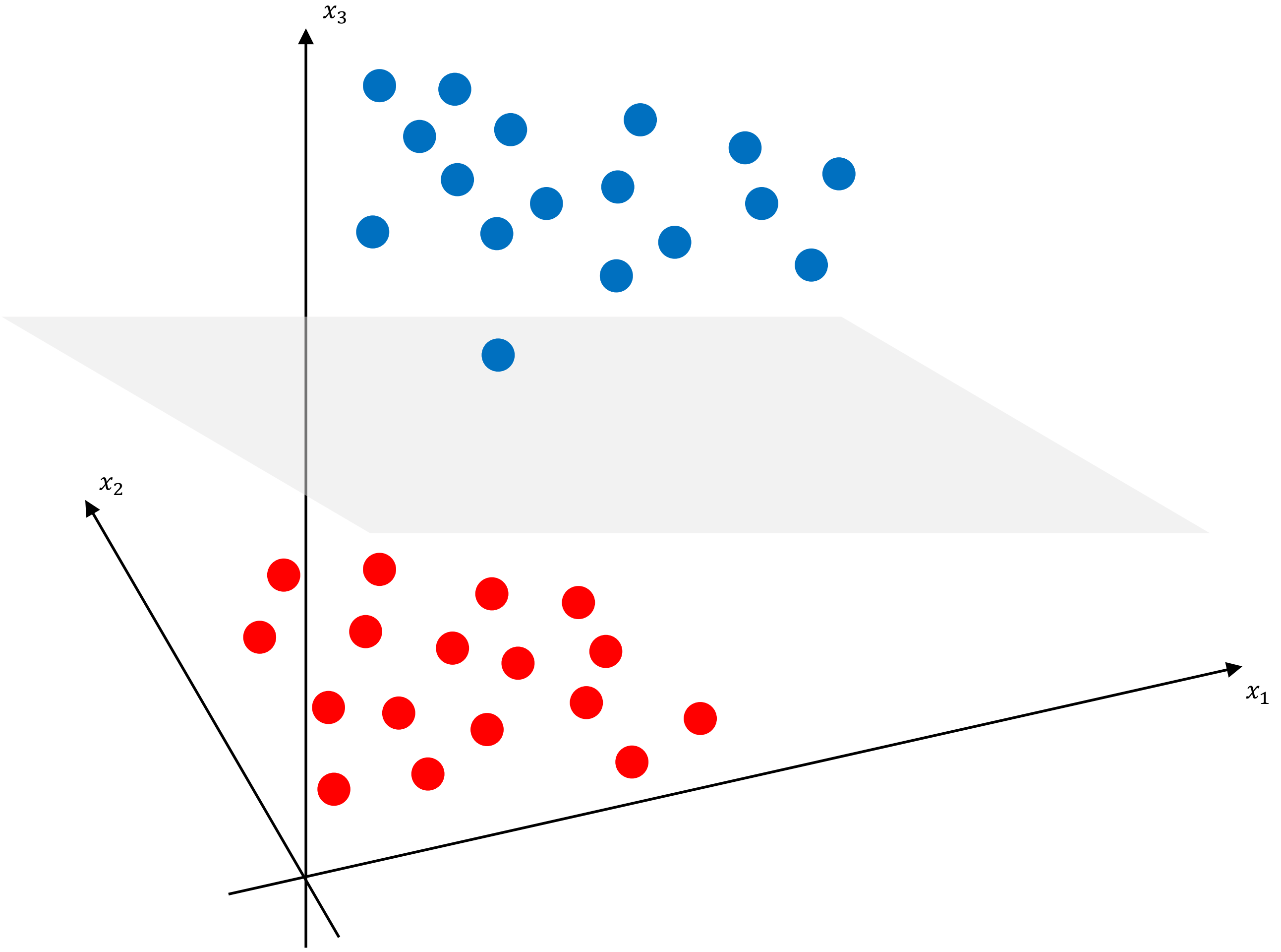
  We need to choose the tradeoff parameter $C > 0$ related to ξ_i.

  Another idea is kernel trick which maps from feature space to higher dimensional space.

  Representative kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ is

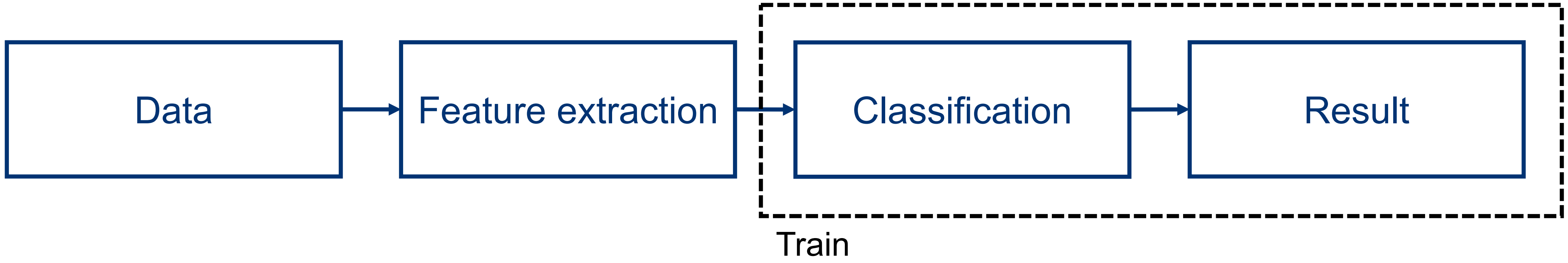
  - RBF(Gaussian): $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2}$.

  We need to choose $\gamma > 0$.



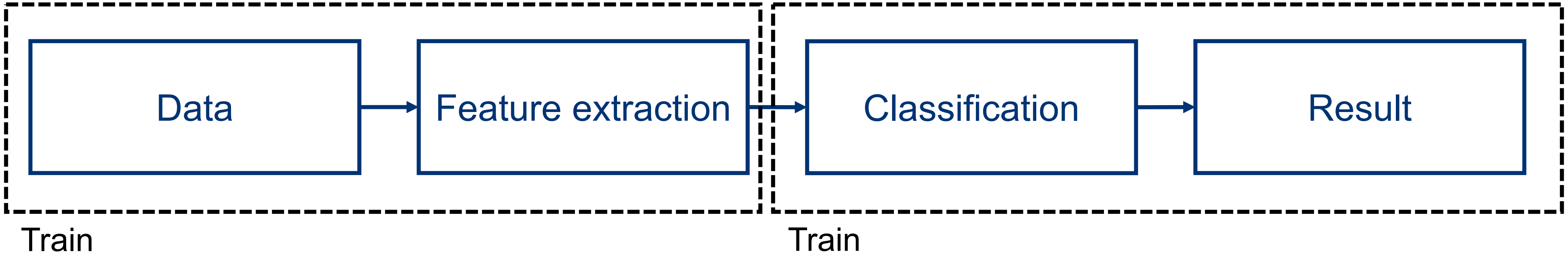
Classical CAD

Extracting meaningful features often results in a loss of good characteristics because it depends heavily on the problem. Therefore, a series of trial-and-error is required to get optimal results which can cause a lot of costs.



Conventional approach generates features by hands (handcrafted features).

Feature learning allows a system to automatically discover the features (non-handcrafted features).



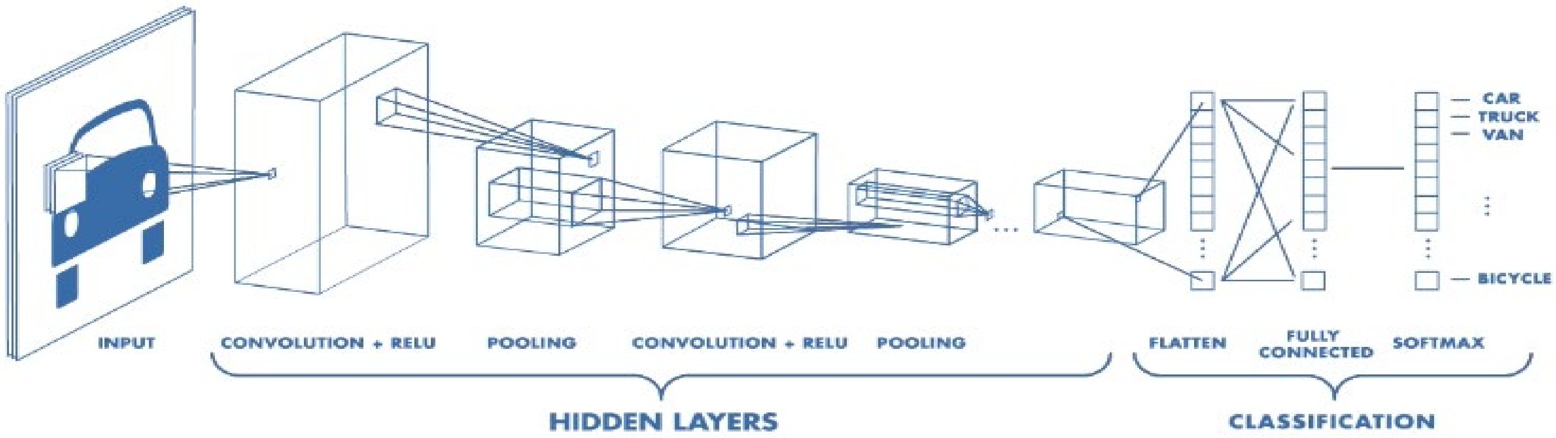
Deep Learning

Deep learning proposes the model that acts on feature learning with domain knowledge and classification at the same time.

Data → Feature extraction → Classification → Result

Train

Consider convolutional neural networks (CNN), one of famous deep learning structures.



CNN example
Patel et al, Introduction to deep learning: what are convolutional neural networks? (MATLAB video, 2017)

Convolutional Layer

- Convolutional layer

  Convolutional layer transforms the 3D input (width x height x depth) to 3D output (width x height x depth).

  Data has various features, so apply multiple filters with small size to catch them.

  - Output of depth

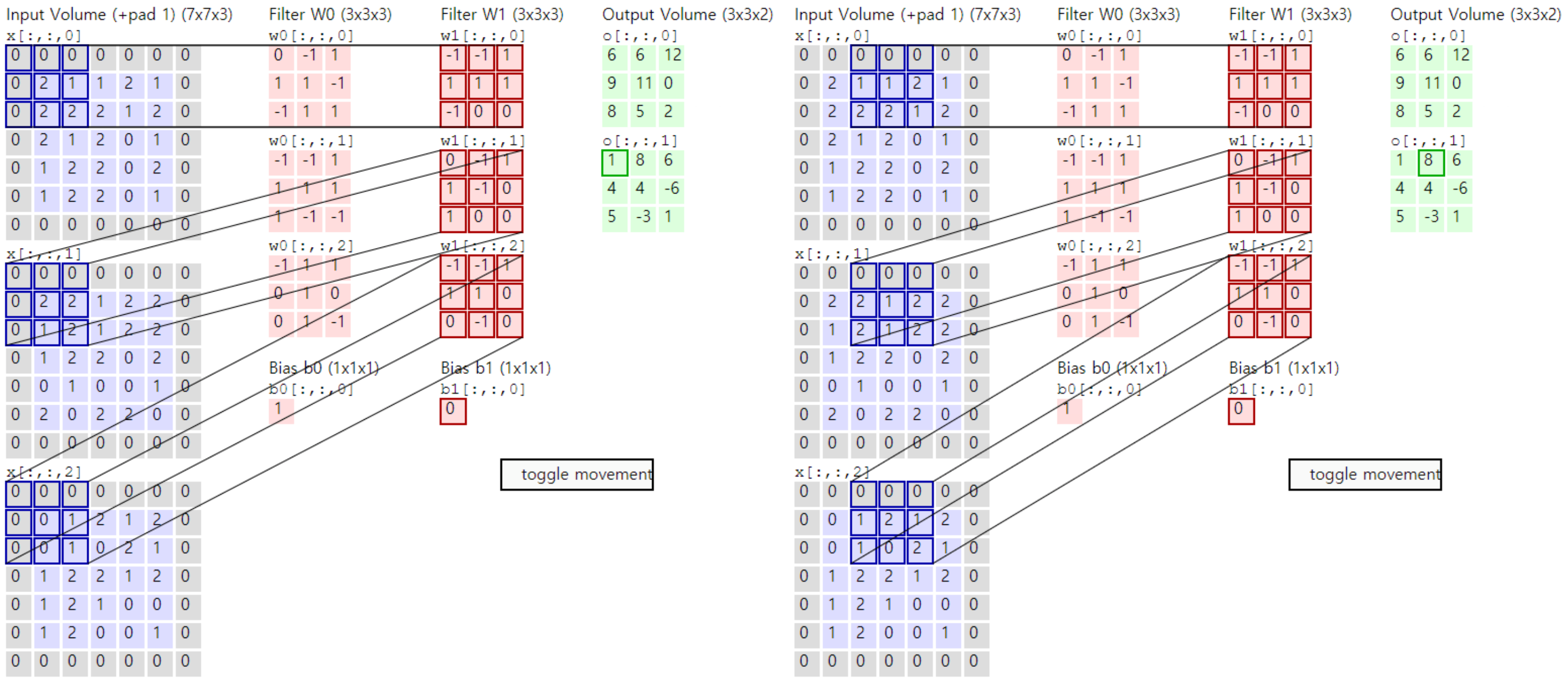
    The number of filters we use.

  - Stride

    Step-size of sliding the filter.

  - Zero-padding

    Padding the input volume with zeros around the boarder.



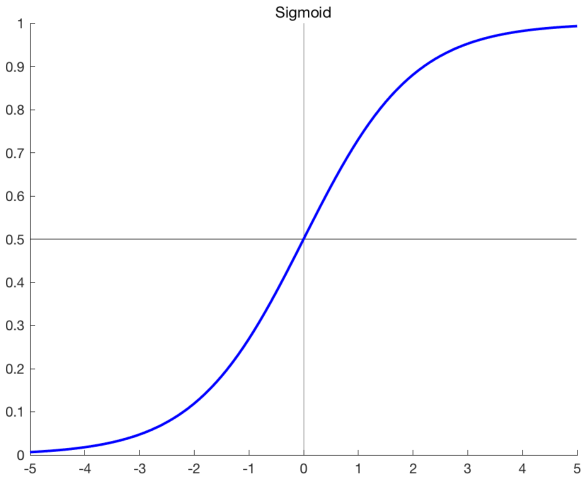
Example Convolutional layer computing.
Input = (6x6x3), zero padding 1, stride 2, and 2 filters. Then output = (3x3x2).
Stanford University, Convolutional neural networks for visual recognition (Lecture, CS231n)

Nonlinear Activation Function

- Nonlinear activation function

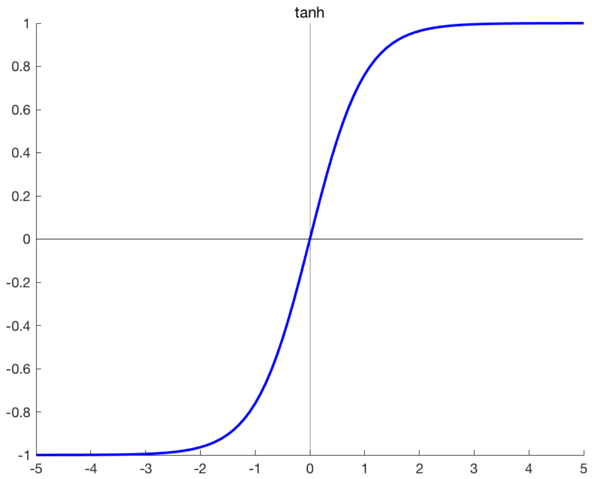
  It gives the filter nonlinearity.

  ReLU(Rectified Linear Unit) is preferred to avoid the gradient vanish so that computation is faster.



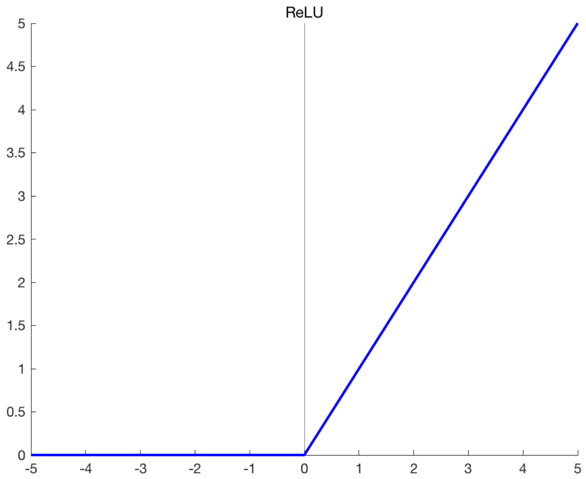
Sigmoid

$$\alpha(x) = \frac{1}{1+e^{-x}}$$



tanh

$$\alpha(x) = \tanh(x)$$



ReLU

$$\alpha(x) = \max(x, 0)$$

6	6	12
9	11	0
8	5	2

1	8	6
4	4	-6
5	-3	1

o (3x3x2)

ReLU →

6	6	12
9	11	0
8	5	2

1	8	6
4	4	0
5	0	1

$\tilde{o}$ (3x3x2)

Example Applying nonlinear activation function.
ReLU is used and $\tilde{o} = \alpha(o)$.

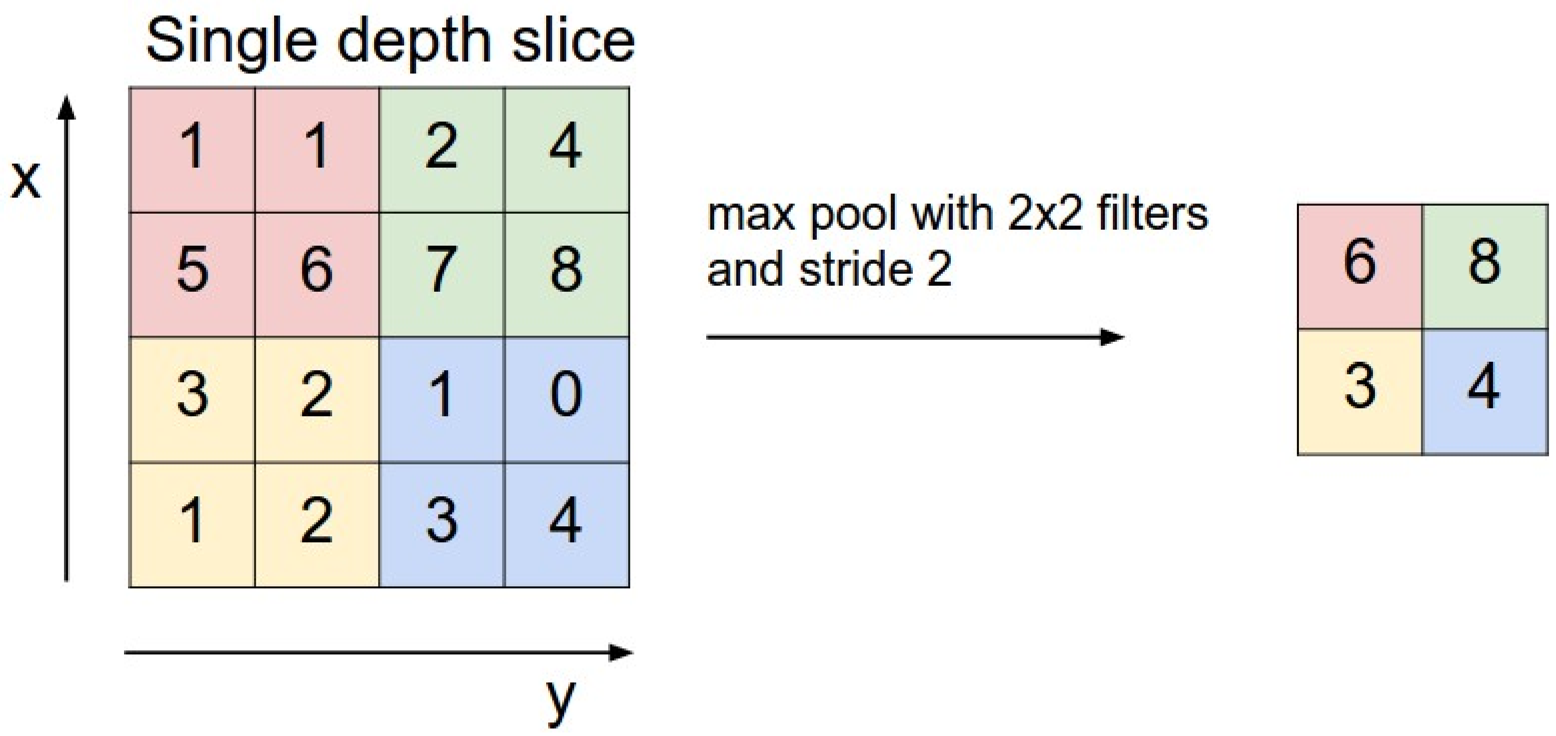
Pooling Layer

- Pooling layer

  Reducing the spatial size(width and height) and to reduce the amount of parameters and computation in the network.
  Max pooling and average pooling are widely used.

  - Stride

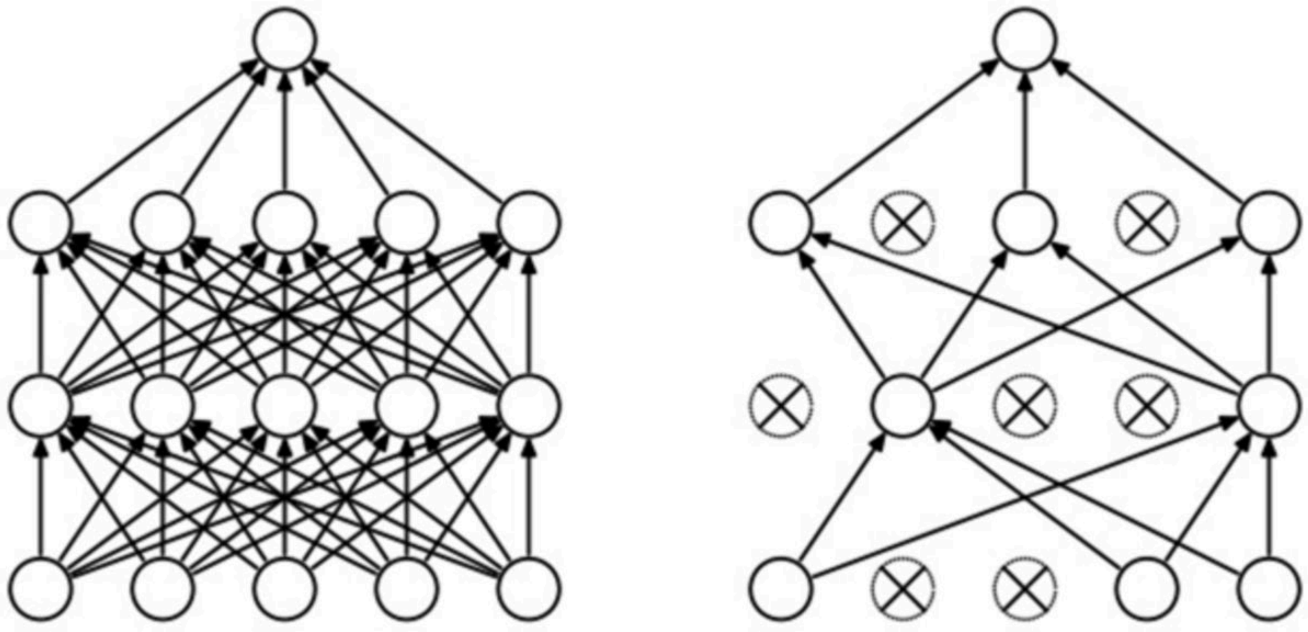
    Step-size of sliding the filter.



Example Pooling layer computing. Input = (4x4x1), stride 2, and max pooling. Then output = (2x2x1).
Stanford University, Convolutional neural networks for visual recognition (Lecture, CS231n)

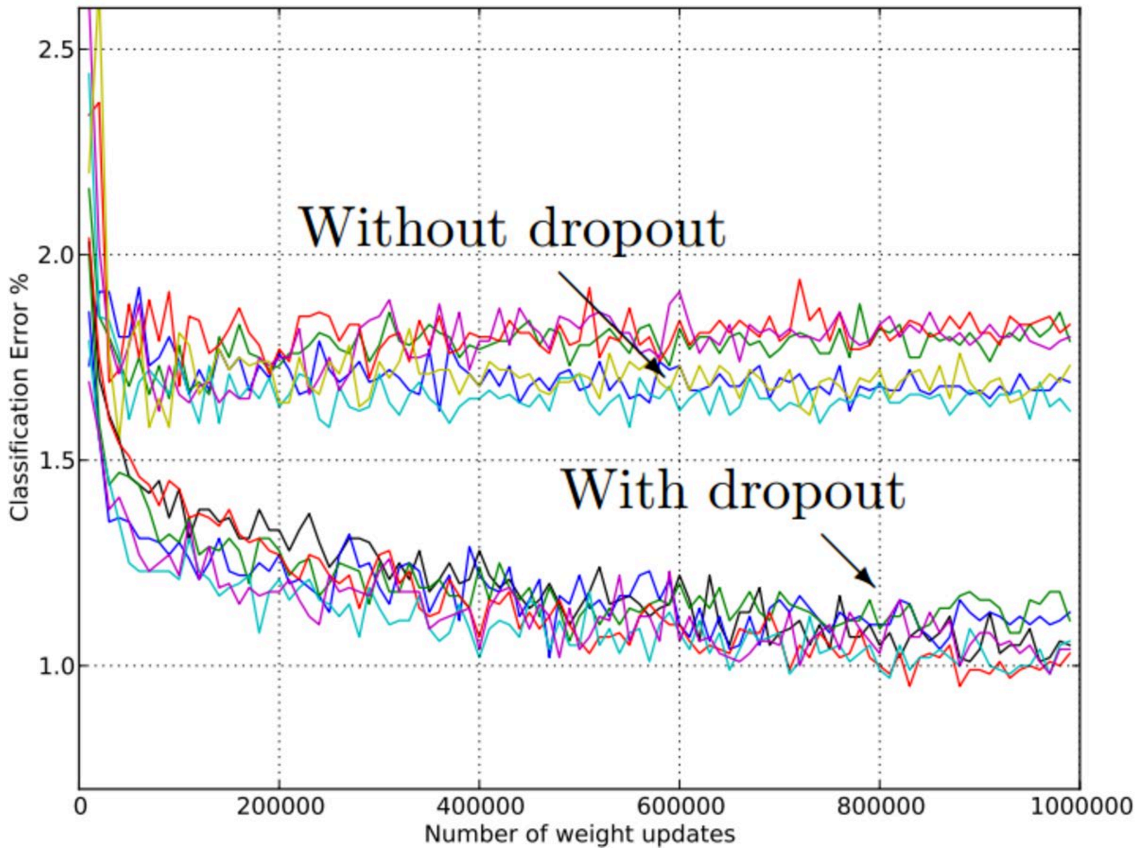
Drop out layer

Drop out



(a) Standard Neural Net

(b) After applying dropout.



Fully-Connected Layer

- Fully-connected layer

  Full connections to previous layer and it acts on the classifier.

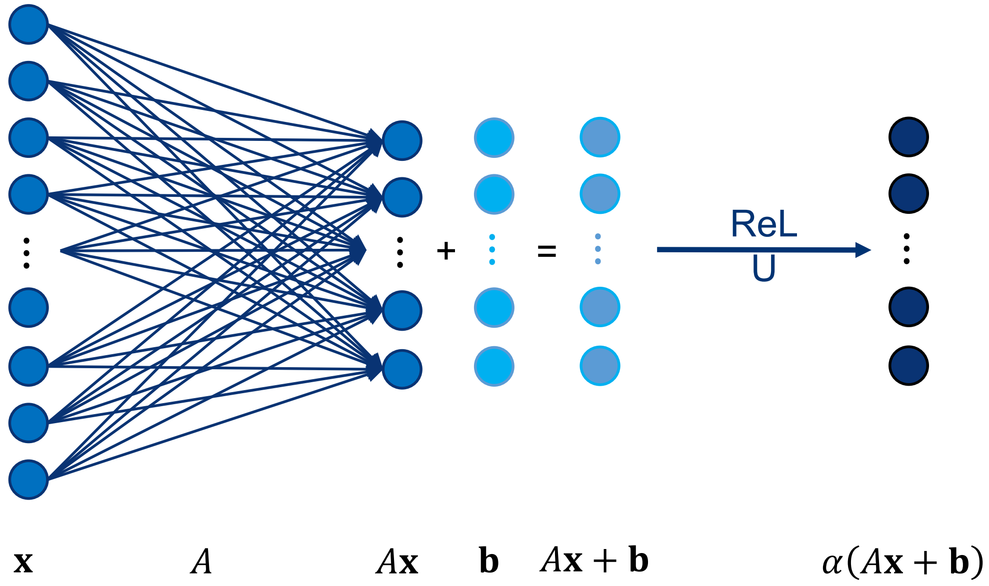
  Input is (1 x 1 x depth) so that it can be considered as a vector.

  Nonlinear activation is applied after fully-connected layer except the last full-connected layer.

  After the last fully-connected layer, softmax function is applied to compute the class scores.

  - Softmax function

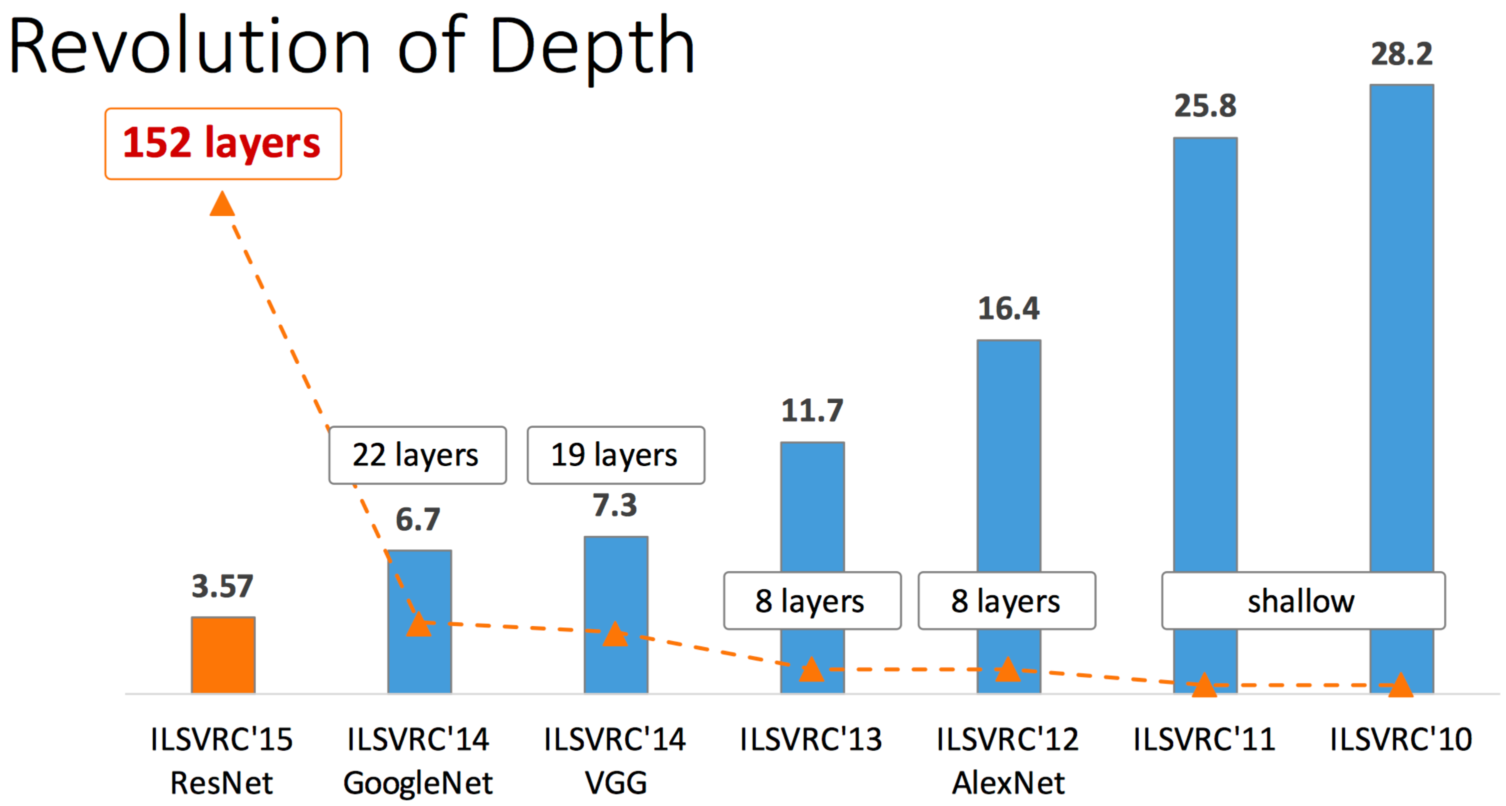
$$\mathrm{softmax}(x_i) = \frac{e^{x_i}}{\sum_{i=1}^{N} e^{x_i}}.$$



Example Fully-connected layer computing and applying nonlinear activation. Input $\mathbf{x}$ = (m x 1), connection A = (n x m), bias $\mathbf{b}$ = (n x 1), and ReLU α. Then output $\alpha(A\mathbf{x} + \mathbf{b})$ = (nx1).

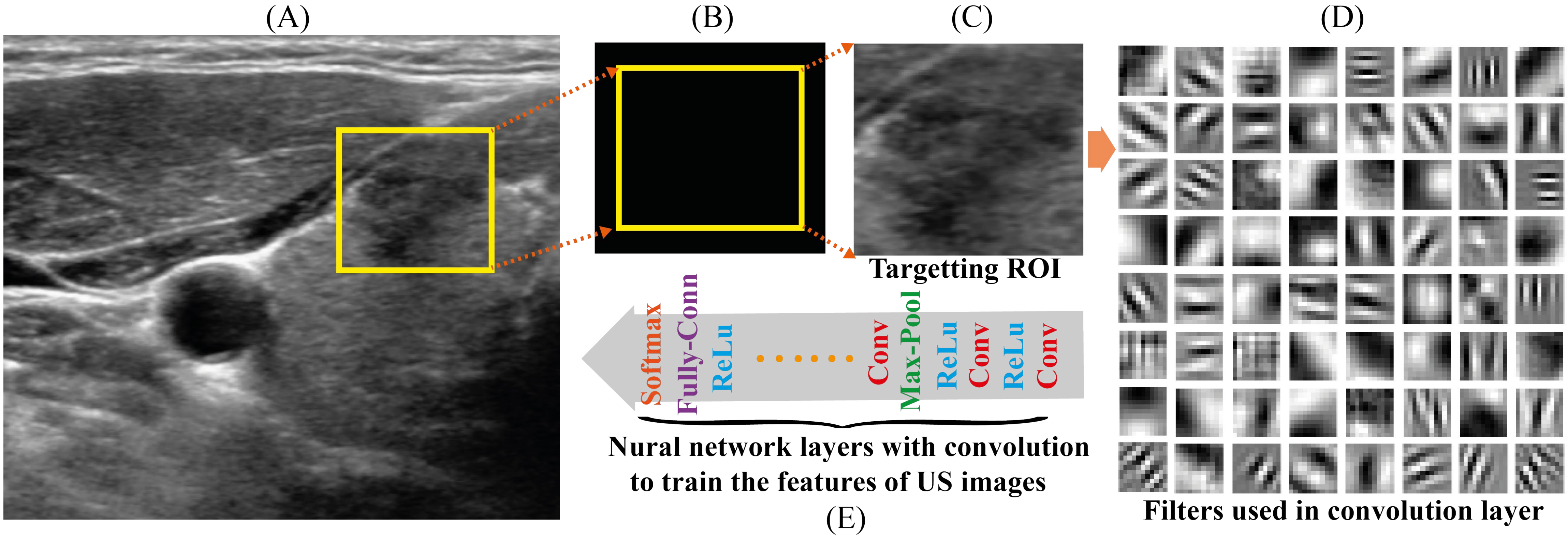
Outstanding Results of CNN

- ILSVRC = ImageNet Large Scale Visual Recognition Competition
  almost 1,200,000 data with 1000 classes
- In ILSVRC, CNN first introduced in 2012 and the performance is outstanding.
- Development using CNN is faster.



ILSVRC results from 2010 to 2015.
He et al, Deep residual learning for Image recognition (2016)

Feature Extraction – overall process

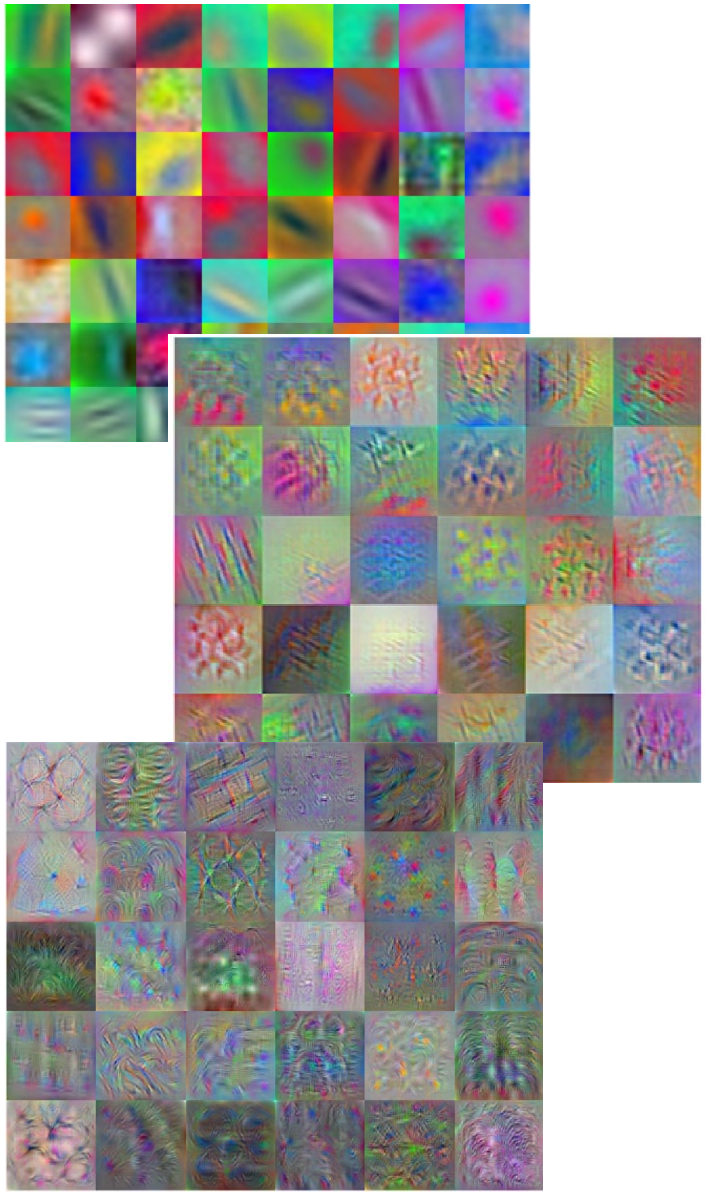


Example : AlexNet – Feature Extraction

Conv1
Relu1
Norm1
Pool1

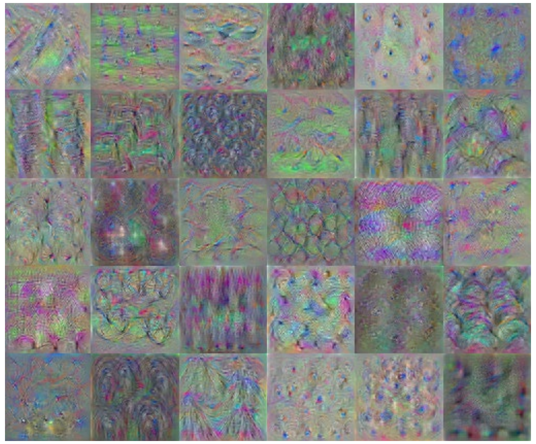
Conv2
Relu2
Norm2
Pool2

Conv3
Relu3



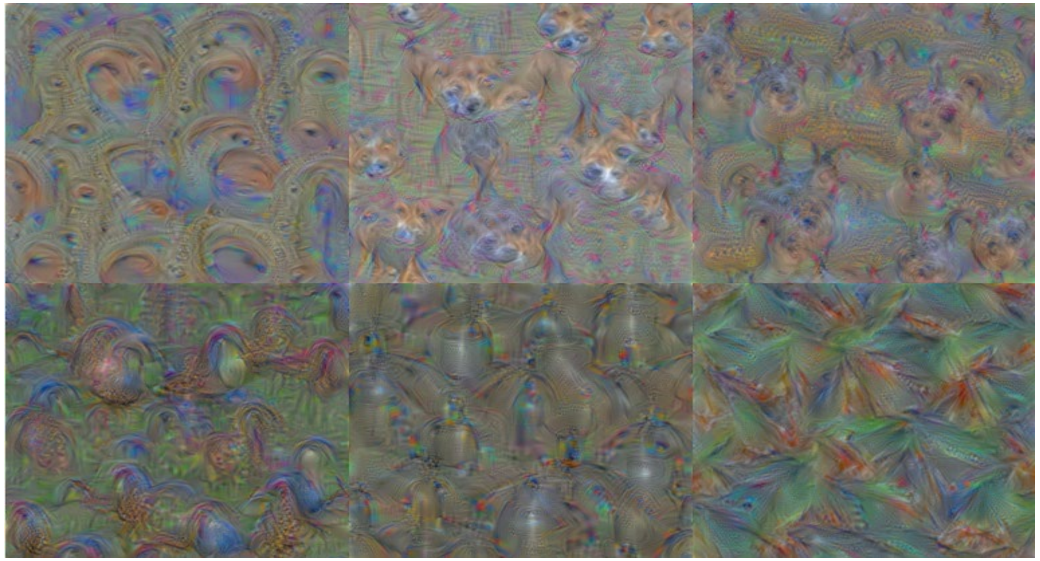
Conv4
Relu4

Conv5
Relu5
Pool5

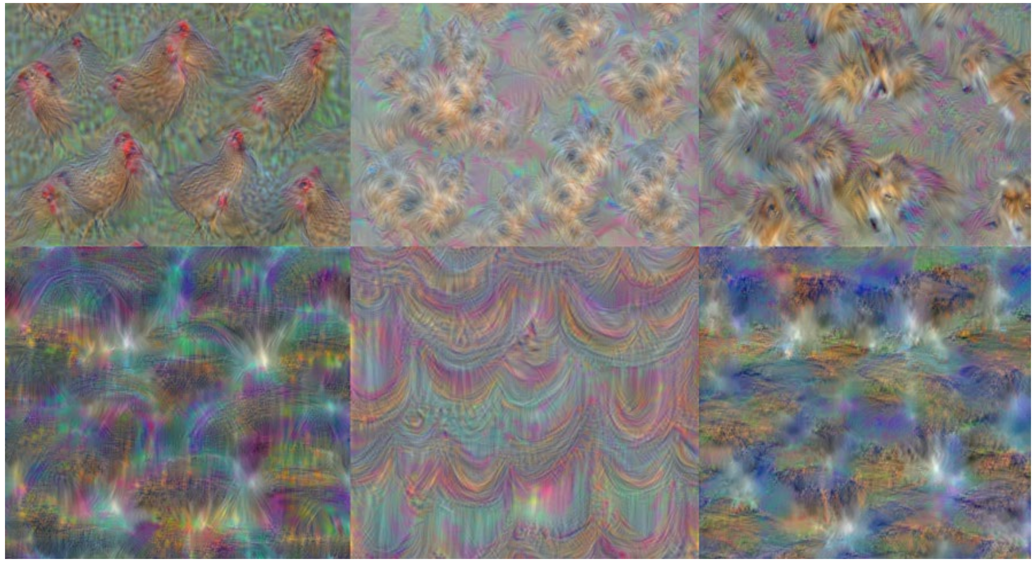


Example : AlexNet – Feature Extraction

Fc6 - Relu6 - Drop6

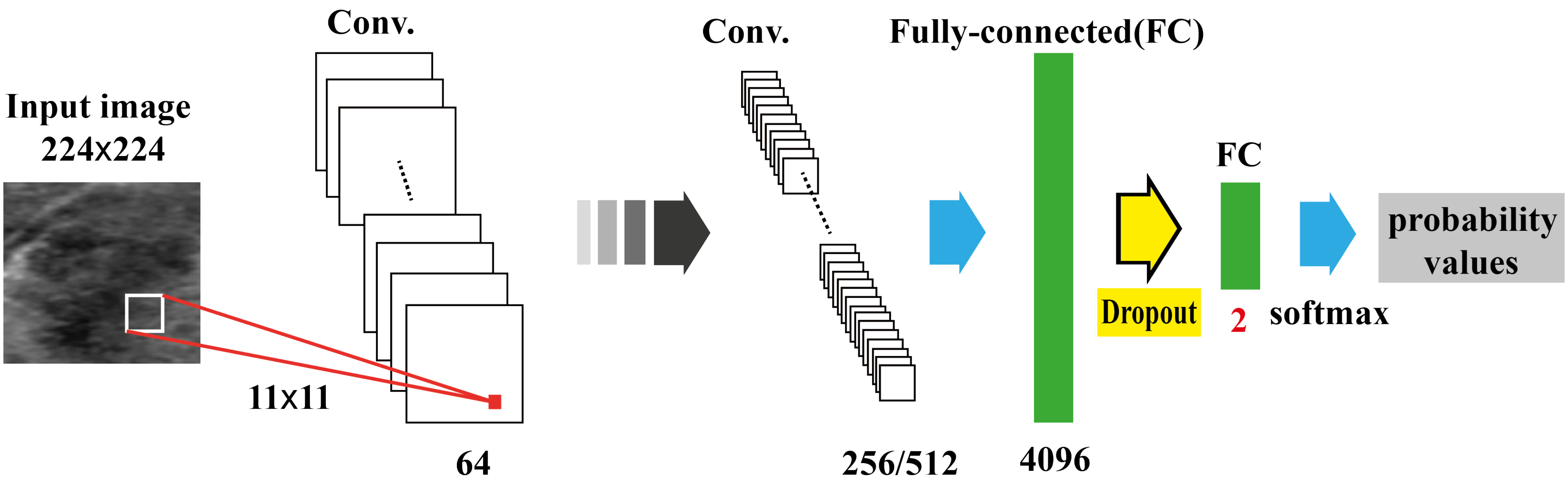


Fc8 - Softmax



Fc7 - Relu7 - Drop7

Feature Extraction – overall process



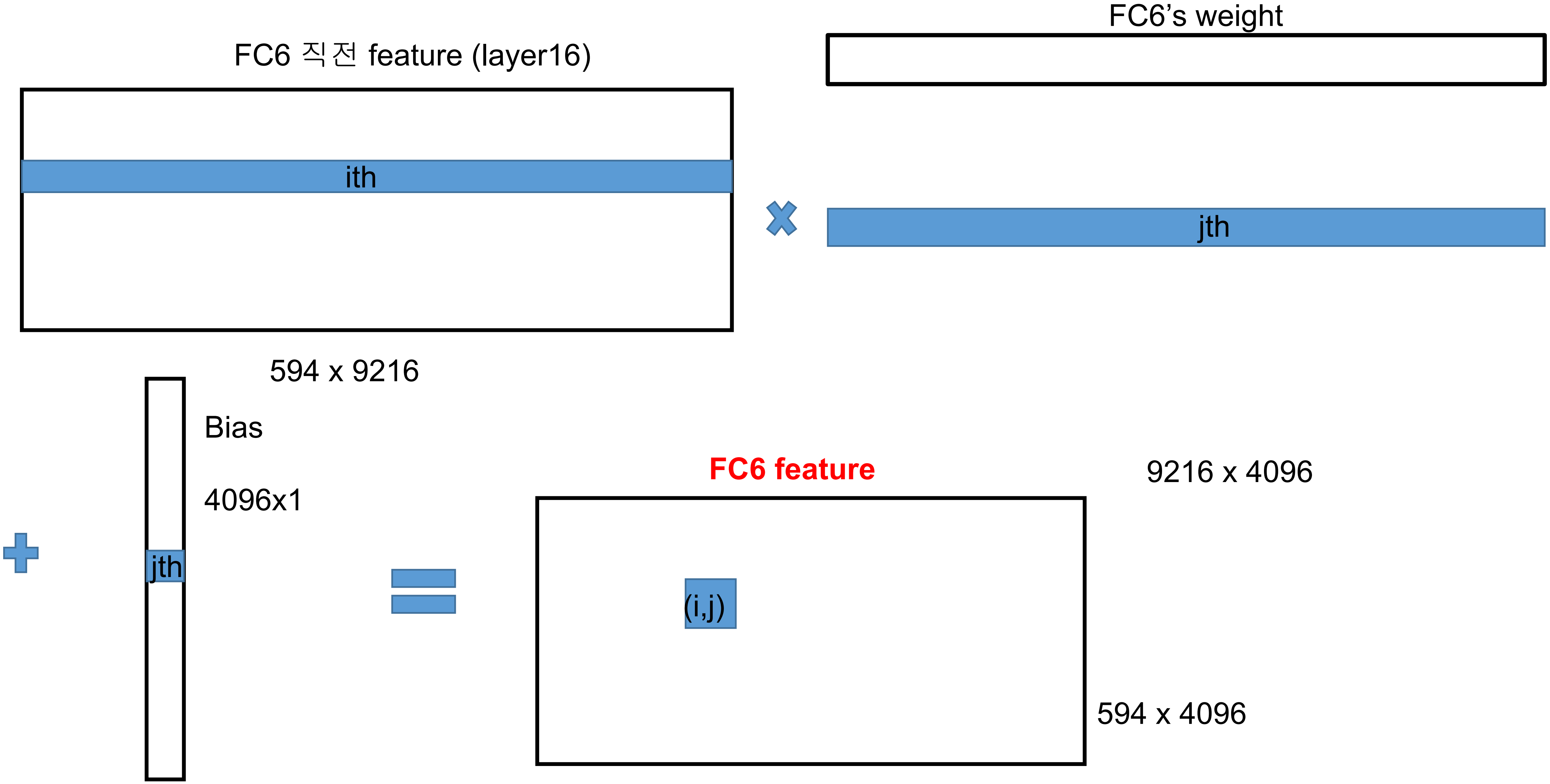
Example : AlexNet – Feature Extraction

[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

Train image: 594

6x6x256 = 9216

Example : AlexNet – Feature Extraction



Example : AlexNet – Feature Extraction

FC6 Features

594x4096 single

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	-10.0524	-7.3053	-10.0279	-5.9850	-1.1304	-2.2774	-8.9562	-2.0188	4.2996	-3.0799	-2.6134	-6.3159	-7.2008	0.0529	-6.6125	-1.6476	-9.6652
2	-12.6945	-4.6964	-4.7964	-6.0794	-2.5154	3.6560	-11.4746	-4.4790	-5.9940	-1.2060	2.7607	-1.2645	0.4807	-1.6525	1.6081	4.5932	-7.2904
3	-12.8434	-3.8693	-7.1252	-16.4763	-6.0867	-2.5018	-12.1147	-5.5396	-5.0504	-8.4952	-0.6999	-10.1645	-5.4002	0.3029	-10.5533	-9.7168	-14.1299
4	-10.6680	-10.3614	-11.5492	-8.5896	-3.2800	-2.3712	-12.0043	-5.2355	-4.3763	-5.2568	0.5840	-2.3008	-0.9851	-3.3943	0.3492	-0.0562	-13.2324
5	-9.4853	-2.7451	-5.8542	-7.0851	-1.5814	3.0937	-12.9630	-6.1883	-1.7918	3.2978	5.5857	1.1986	-0.3680	-5.3279	-0.2068	4.9027	-10.3189
6	-9.6521	-4.6634	-4.6361	-3.3165	1.2386	6.2837	-12.0828	-7.2156	-2.2693	-0.7450	5.4057	-7.3945	1.4167	-0.5443	-1.9530	2.4427	-7.2184
7	-11.2315	-8.7407	-7.0433	-0.1484	4.5832	-0.0875	-10.3266	-8.4115	-0.5152	-4.4539	4.3578	-7.1955	-8.8385	-8.3253	1.4729	-2.7571	-11.7381
8	-8.6928	-5.0548	-4.3843	0.3110	1.4943	5.2356	-6.1409	-6.1843	7.5856	0.8224	7.9765	-10.5443	-1.5775	-1.9113	-3.0277	0.2646	-5.0912
9	-8.2086	-6.1527	-6.4176	-0.2672	1.0193	6.2592	-10.1538	-6.7400	4.3408	2.4601	5.3555	-11.8475	0.3163	-1.3523	-4.9409	6.4918	-7.6730
10	-4.2627	-4.3079	-6.3708	-8.4616	-2.1247	3.6830	-9.6782	-4.5905	2.5772	-3.4952	-0.2469	-1.7125	0.0045	-9.2431	1.1432	-2.3397	-6.8485
11	-11.0697	-9.1591	-2.8381	-13.5645	-1.5675	-2.0480	-10.3500	-7.1333	-0.4048	-5.5405	0.1110	-10.0292	-5.2349	1.6533	-7.2150	-7.0371	-12.7210
12	7.9365	2.8335	5.0096	4.6175	0.5109	2.9364	10.6335	3.8335	0.8121	5.1464	3.1737	6.7110	7.2781	3.6418	0.9198	0.9671	10.3043
13	-11.3102	-13.5841	-8.2198	-11.7206	-5.1626	-11.1110	-11.4773	-7.1056	-6.0481	3.2913	6.8043	0.0873	-5.7940	5.2007	-8.9709	-15.5002	-14.4938
14	-7.8656	-5.0375	-11.4900	-18.2956	-2.2875	-5.5520	-10.4198	-4.2955	-3.0429	-3.7156	0.5927	-4.4298	-1.6833	4.2500	-9.6159	-6.4494	-11.6674
15	-9.7864	-2.4722	-4.4094	1.8789	3.4579	3.6574	-9.9743	-6.9234	-0.0389	-0.0558	6.3875	-7.6009	2.6833	0.2722	0.9535	2.6285	-6.1058
16	-7.4317	-5.6857	-7.9149	4.6885	6.2439	7.8335	-7.9817	-6.9426	3.6581	0.1099	7.0851	-6.0560	-1.1419	-0.0128	0.7669	6.5825	-9.9803
17	-9.5884	-8.0933	-8.8117	3.3384	2.1149	2.2117	-8.3343	-11.2434	3.7444	-3.6611	5.7198	-4.7155	-0.7275	-0.3561	0.0475	3.6237	-13.6119
18	-8.5365	-4.3180	-3.8605	1.7503	-0.0925	-2.9494	-9.0258	-7.2710	3.0778	2.3694	4.4537	-5.5055	-3.0503	-5.8340	-5.7727	-0.4316	-8.8289
19	-10.6666	-5.2921	-3.5396	-2.4228	-0.1511	1.2619	-9.1544	-7.9685	-2.5432	-3.0850	4.6193	-3.9584	0.5641	1.7539	1.4376	-1.6685	-11.0098
20	-9.9551	-5.5355	-8.4123	-6.5886	-2.0140	-2.0883	-9.3802	-6.2586	3.2467	-4.5648	2.9441	-6.1910	-1.4518	-0.4068	-3.5765	-3.9447	-10.6259
21	-8.8158	-4.1876	-5.0227	-7.2081	-0.9771	4.6588	-8.5589	-4.4808	-0.7624	-0.5423	5.1920	-3.6195	0.6186	-5.4550	-1.2937	2.2552	-7.9696
22	-7.2221	-4.8274	-6.6797	-1.0075	1.2572	-1.3295	-12.5832	-7.7488	2.9054	0.4058	5.6602	-5.6424	0.1422	-4.0743	-1.8597	2.0908	-8.0480

Example : AlexNet – Feature Extraction

FC7 Features

594x4096 single

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
49	-4.6883	-2.6209	1.5288	1.5505	-2.2454	-1.8788	-5.2284	0.8465	-1.0774	-2.2381	-1.9902	-1.7848	-6.6574	-1.1343	-2.5492	-3.1782	-2.7998
50	-4.7801	-1.8624	0.5818	0.4723	-3.2671	-2.2711	-5.6901	0.0880	-2.1856	-0.7124	-1.8565	1.5895	-8.3259	-0.8175	-2.4487	-1.6805	-2.0786
51	-4.6571	-3.1362	-0.4551	1.1888	-2.3157	-2.2419	-5.5302	1.3509	-1.6518	-2.4192	-1.8302	1.6237	-7.5630	-0.6131	-2.9331	-4.8671	-1.9731
52	-3.0186	-3.2790	1.1050	0.2592	-1.9135	-1.7730	-5.4676	1.8891	0.2701	-2.3524	-2.3155	-2.1388	-9.2003	-1.4011	-3.9992	-3.6775	-2.2545
53	-1.1307	-1.2343	1.7185	0.6677	-2.7430	-0.7016	-3.6164	-0.9341	-0.9177	-1.5608	-1.6971	0.7859	-8.6417	-0.3882	-4.7683	-0.4859	-3.1078
54	-3.3725	0.1059	3.0717	1.4900	-2.1481	-0.9503	-2.9874	-0.4081	-1.8770	-2.3047	-3.0683	0.9692	-6.2255	1.0191	-2.2729	-2.1684	-1.7941
55	-4.1872	0.1905	2.3742	1.3918	-1.1684	-1.0737	-2.4628	-1.5443	0.6584	-2.0975	-2.7386	0.8334	-7.4657	0.4198	-1.8999	-0.6900	-2.0905
56	-6.3744	-2.3352	1.4426	-1.0553	-2.1845	-4.3706	-5.5064	-1.0890	-4.5944	-0.2464	-3.6421	3.5497	-9.0293	-1.2407	-2.4798	-2.3570	-5.0450
57	-3.3933	-1.8376	3.2381	-1.4066	0.2177	-1.9272	-3.8339	1.3252	-1.9915	-1.7664	-0.7051	-2.4484	-5.9229	0.4111	-2.9836	-1.9567	-0.7303
58	-4.1154	-3.8089	0.3644	1.5528	-2.0896	-1.9433	-6.9603	-0.7116	-1.6621	-1.4116	-3.0963	1.7629	-9.4549	-2.0749	-3.8017	-2.6437	-3.9228
59	-4.0019	-0.8798	2.0669	0.8743	-2.4778	-1.1915	-4.5024	0.4930	-0.3493	-1.0908	-2.8739	0.5911	-8.4356	-0.0788	-3.2444	-1.7428	-3.2241
60	-2.0538	-3.0819	2.7041	-0.4126	-2.5229	-0.8368	-5.4017	1.1253	0.1040	-2.3910	-1.6025	-1.1352	-6.3086	-0.8491	-1.3985	-3.0836	-1.2915
61	-2.6685	-1.3877	2.5823	-0.2076	-1.2596	-2.0287	-5.1626	-0.1934	-0.2218	-1.4087	-1.2702	0.3543	-6.5492	0.4424	-1.1340	-0.7195	-2.5388
62	-2.7659	-2.0143	2.3064	2.2973	-1.7235	-1.1161	-4.8267	0.6877	-1.0023	-2.4131	-3.2939	0.1010	-7.4398	-1.7334	-3.8121	-3.8246	-1.6842
63	-3.7985	-2.1073	2.0244	2.4896	-1.0844	-2.1125	-5.9992	0.2640	-2.2666	-1.5039	-1.9154	2.2370	-7.5200	-0.9913	-2.3294	-1.0922	-3.0830
64	-4.8162	-2.7466	1.1129	2.2409	-1.3494	-1.2011	-5.2651	-0.4621	-1.4928	-1.3814	-2.3031	1.0943	-7.0032	0.4499	-0.7397	-0.9253	-4.1918
65	-3.1527	-1.5272	2.6154	1.3367	-0.5596	-2.0247	-4.8508	1.7031	-2.1729	-1.1982	-1.8204	-0.7007	-7.5433	-0.4120	-3.3690	-1.7475	-2.4611
66	-5.4703	-1.6861	0.0650	0.8265	-1.2702	-2.9783	-5.1866	0.5127	-3.0288	-1.3621	-3.4347	1.8885	-8.8669	-0.3171	-2.2018	-2.5670	-3.9579
67	-4.9264	-2.1536	2.6325	1.7408	-1.4197	-1.5483	-5.9990	-1.3549	-1.6383	-0.9280	-2.9523	2.3862	-7.2531	-0.2729	-3.9437	-3.0390	-3.5674
68	-6.4155	-2.5474	0.3580	0.4339	-1.7091	-2.9187	-5.6149	-0.1795	-3.8236	-1.6318	-3.3856	3.1248	-8.7770	-0.2987	-3.1038	-2.3098	-4.8905
69	-5.3020	-3.0861	0.3074	0.9462	-2.2428	-1.6282	-3.7811	1.6435	-2.1459	-0.7879	-0.9699	1.7060	-6.5744	-3.2842	-2.4355	-1.4592	-2.6439
70	-2.8415	-2.5114	2.0822	-0.4998	-2.0422	-1.7433	-6.4495	-0.1937	1.5666	-1.1711	-2.0628	1.5488	-6.9607	0.3079	-2.9166	-2.4824	-3.2849

Example : AlexNet – Feature Extraction

FC8 Features

594x1000 single

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0.6772	-2.7984	0.6396	0.6195	0.0487	1.0616	0.7476	-2.1878	-1.4255	1.7664	-0.8377	-3.6805	-2.2573	-0.7124	-3.4645	-1.3709	-4.1907
2	-0.1629	-1.3490	-1.0225	0.4584	-0.0613	3.7071	1.0612	-3.2916	-1.8554	-2.2365	-0.0326	-3.0064	-1.6887	0.0313	-3.4046	-0.8696	-3.9588
3	-0.3303	-3.6537	1.6302	0.7343	-0.4033	1.9997	0.8008	-1.0541	-0.8818	3.4582	-1.5449	-4.2534	-3.1313	-1.9201	-3.2175	-2.8394	-4.0686
4	0.3560	-1.5093	0.7965	1.3312	0.2140	5.2392	2.2962	-3.0074	-1.8411	-0.4090	-0.4059	-4.1327	-1.4442	1.1654	-3.3186	-1.3701	-3.6144
5	1.2022	-1.4545	0.5672	1.1469	0.4639	2.6548	1.6637	-3.7688	-1.8251	0.1351	1.0684	-2.8017	-0.0955	0.7546	-2.7960	0.5235	-2.9044
6	-0.3809	-2.9085	1.1381	1.8129	0.9058	1.6350	1.2397	-3.3758	-2.8921	-0.2930	-0.6286	-2.8286	-1.5832	-0.4346	-3.7926	-1.4105	-4.4738
7	3.0426	-1.2155	5.6820	4.9141	4.0943	4.1703	3.3394	-4.4438	-3.8466	-2.2694	-0.1738	-3.8677	-1.8711	0.4557	-3.2238	-0.7820	-4.4999
8	1.7880	-1.8028	2.1588	1.4915	2.1676	1.5407	1.3514	-4.2407	-3.1595	-1.2009	-0.7960	-3.8811	-2.4925	-0.8032	-4.0026	-1.2345	-5.5035
9	1.2305	-2.1736	2.9047	1.6176	2.1908	1.9532	1.9715	-3.5814	-2.6869	-2.0106	-0.7126	-3.7994	-2.0572	-0.0849	-4.2628	-0.7036	-4.3128
10	0.1131	-3.2891	0.4780	1.6599	1.3511	0.8343	0.2476	-3.8728	-2.8042	-1.7560	-1.4151	-3.5736	-2.8106	-1.1278	-4.5425	-2.3575	-5.0849
11	1.6027	-1.9045	3.6447	1.4541	1.4220	3.4532	2.3772	-2.8330	-2.0032	-0.5946	-1.3384	-3.3893	-3.1229	-1.8650	-4.1669	-2.6108	-4.6760
12	0.6554	-2.2162	1.8799	2.1474	2.3553	0.3049	1.4705	-2.4223	-1.8673	0.9924	-1.1680	-3.1563	-1.3847	-0.3348	-3.3761	-1.3925	-4.1143
13	0.8659	-2.3168	2.2935	0.7944	-1.2493	4.5448	2.3244	-2.3716	-1.9694	1.3141	-1.5224	-3.8622	-1.8077	0.2561	-2.9338	-1.9922	-4.1382
14	1.6015	-1.4996	4.4494	1.4303	1.9799	2.7133	3.0224	-1.1448	-1.5092	-0.7038	-0.8948	-3.6424	-2.3989	-0.2355	-3.8475	-1.1138	-3.6389
15	0.4559	-2.3445	2.1884	2.2266	1.7817	2.7813	2.4240	-3.3994	-2.4756	-0.6284	-1.0316	-3.9111	-1.9052	0.5832	-4.5192	-0.7132	-4.6409
16	0.1940	-2.2519	1.1021	0.4232	2.0404	1.7810	1.6347	-2.9682	-2.2078	-1.0598	-0.7517	-3.8903	-1.5403	0.9673	-3.2360	-0.0174	-3.7593
17	1.7357	-2.4331	3.0468	2.4013	2.5824	4.1919	4.0791	-3.9207	-2.9133	-1.3249	-0.9666	-4.4224	-1.8136	1.4477	-3.8483	-1.0061	-5.0681
18	1.6989	-2.1575	2.0083	0.0153	0.7504	2.1248	1.4263	-3.3006	-2.7635	-0.6778	-1.8011	-4.5059	-2.1471	-0.7672	-3.6656	-1.6877	-5.2874
19	1.0387	-1.4788	1.9489	1.8730	2.0684	3.0615	2.9280	-2.8588	-1.9969	-1.3101	-0.5779	-3.4247	-1.4896	0.9042	-3.4535	-1.3542	-4.1778
20	1.3161	-1.4346	5.2571	3.2771	3.8218	3.3975	3.2177	-1.4806	-1.7936	0.2792	0.2542	-3.7378	-1.2942	0.8114	-3.8741	-0.9022	-3.4509
21	0.8455	-1.6468	0.3183	1.4129	1.4421	2.8695	1.1195	-3.0973	-1.7900	-0.2494	-0.7450	-3.4600	-2.2536	-0.9570	-3.7315	-1.1524	-4.7386
22	2.8529	-1.8884	2.9976	2.8418	2.7918	3.9800	2.9642	-4.5473	-3.5805	-2.2142	-0.7829	-4.2885	-1.8753	0.2981	-3.7720	-0.7226	-4.7393

Transfer Learning and Fine-Tuning

- Good learning process requires big data in deep learning.
- In medical imaging area, it is often not available.
- Instead, use CNN models trained by huge amounts of data with various classes (ex. ImageNet).

Transfer learning

1. Remove the last fully-connected layer.
2. Add the fully-connected layer with output = number of our classes.
3. Pre-trained weights are fixed and train the weights of the last fully-connected layer.

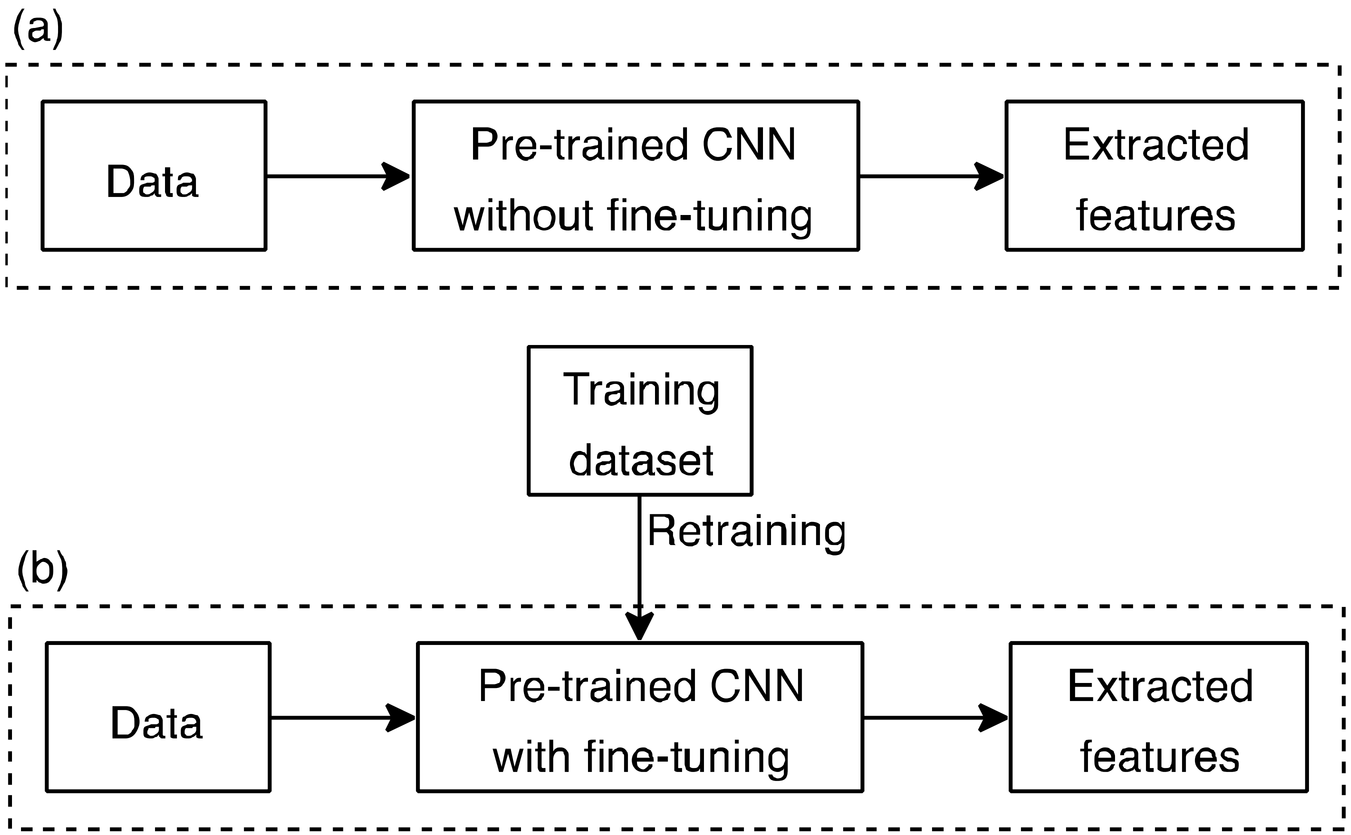
Fine-tuning

1. Remove the last fully-connected layer.
2. Add the fully-connected layer with output = number of our classes.
3. Pre-trained weights are considered as initial information and fine-tune the weights by training dataset.

Sometimes, earlier layers can be fixed.

Feature Extraction from CNN

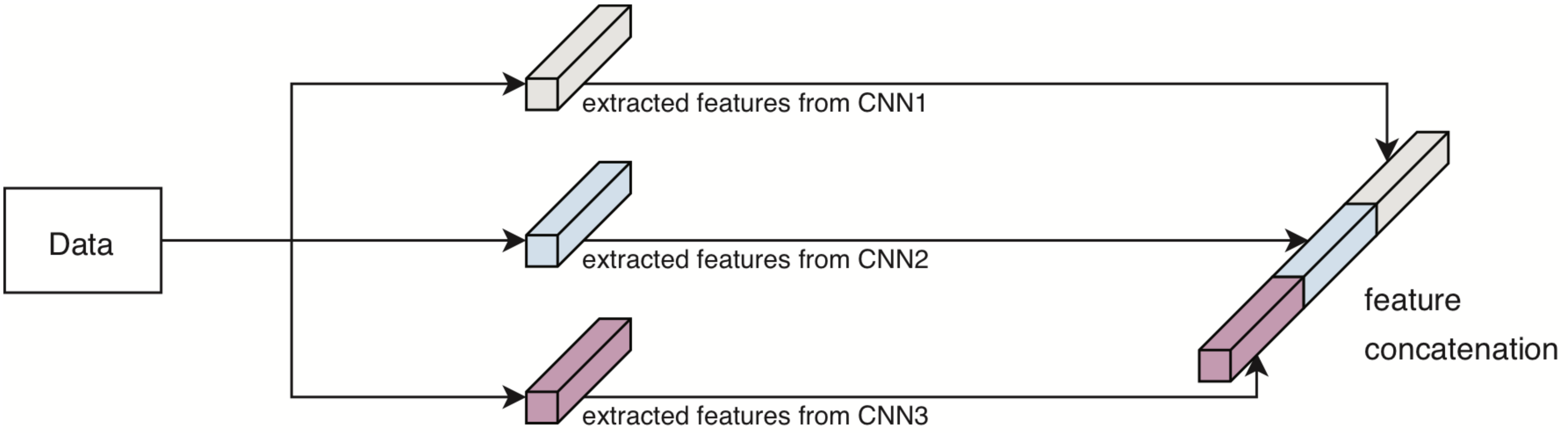
- Two strategies are considered:
  1. Extract features from pre-trained CNN without fine-tuning.
  2. Extract features from pre-trained CNN with fine-tuning.
- For classification, conventional classifiers are used.



Feature extraction from pre-trained CNN without fine-tuning (a) or with fine-tuning (b).

Feature Concatenation

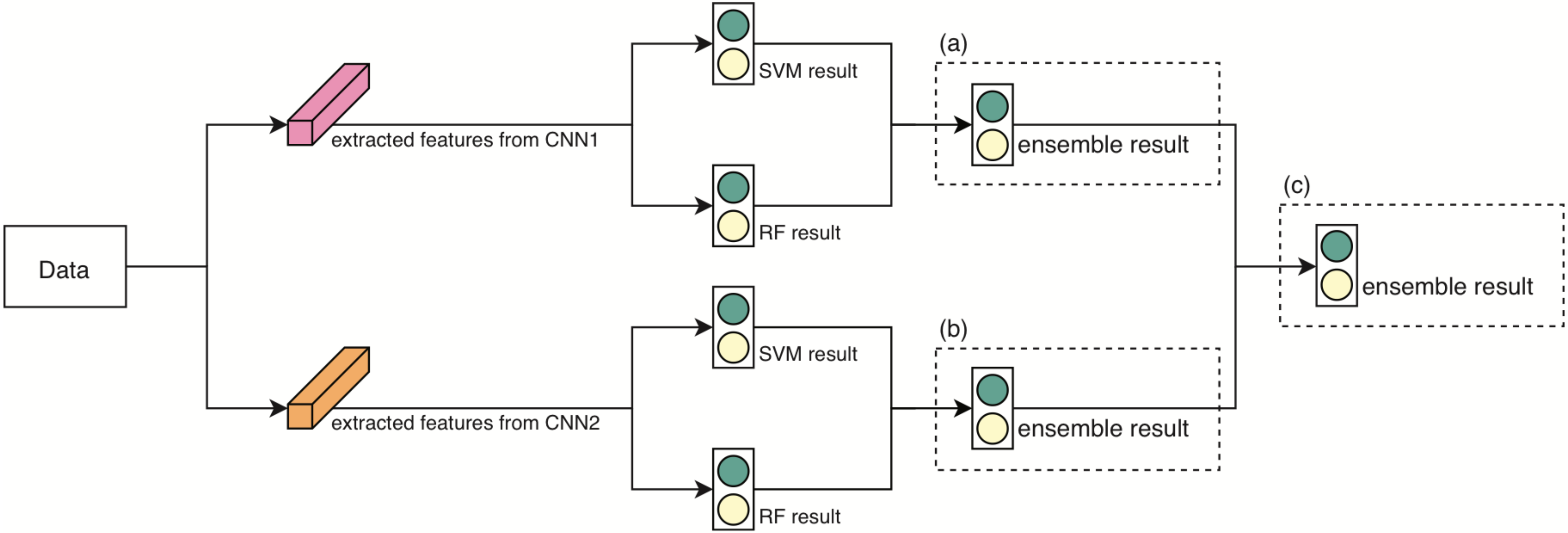
- The features extracted from deeper layer are compressive, so discriminative information may be missed.
- Examine the features extracted from different layers or CNNs in various combinations.



Example Feature concatenation of features extracted from three different CNNs.

Classification Ensemble

- For extracted features, several classifiers are considered such as SVM and random forests(RF).
- To obtain complementary result, apply ensemble method to each classifier result.



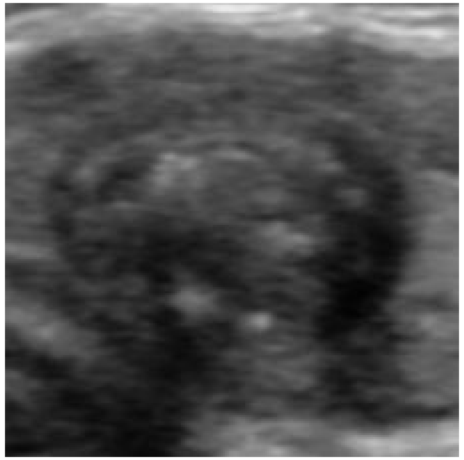
Example Classification ensemble to SVM and RF results from each CNN (a,b) and overall results (c).

Result: Setup

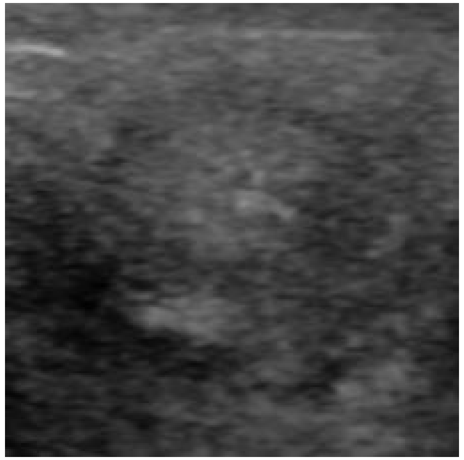
- Dataset: ultrasound images of thyroid nodule by cropping the ROI
  - Labels
    (benign, malignant)
  - Training data: 594 images
    297 benign and 297 malignant
  - Test data: 150 images
    100 benign and 50 malignant
- Performance criteria:
  (1) accuracy (acc) (2) sensitivity (sen) (3) specificity (spe)

$$\mathrm{acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \qquad \mathrm{sen} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad \mathrm{spe} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

  TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative
- Pre-trained CNNs
  1) AlexNet 2) OverFeat-accurate 3) VGG-F
  4) VGG-19 5) ResNet-50 6) Inception-v3
- Classifiers
  1) SVM 2) RF



Example Benign



Example Malignant

Result: Fine-Tuning

- Optimizer: stochastic gradient descent with momentum

  momentum = 0.9, mini-batch = 33, cross-entropy loss function.
- Validation: 6-fold cross-validation
  - to find optimal epochs, learning rate, and learning decay.
  - OverFeat and VGG are not applied yet.
- Extracted layer
  - AlexNet, OverFeat, VGG, VGG-verydeep: 1st and 2nd fully-connected layer (fc1, fc2)
  - ResNet, Inception: average pooling layer (avg)

Net	Language (version)	Package (version)	Pre-trained weights	Epochs	Learning rate	Learning decay
AlexNet	MATLAB (R2018a)	Neural Network Toolbox	Neural Network Toolbox	7	1e-03	0.1 for every 5 epochs
OverFeat	Lua (5.3.2)	Torch7	NYU CILVR lab	20	1e-03	decreasing log-scale to 1e-04
VGG	MATLAB (R2018a)	MatConvNet (1.0-beta25)	MatConvNet	20	1e-03	decreasing log-scale to 1e-04
VGG-verydeep	Python (3.6.5)	Keras(2.1.6), Tensorflow(1.7.0)	Keras	54	1e-05	-
ResNet	Python (3.6.5)	Keras(2.1.6), Tensorflow(1.7.0)	Keras	37	1e-04	-
Inception	Python (3.6.5)	Keras(2.1.6), Tensorflow(1.7.0)	Keras	9	1e-03	0.1 for every 5 epochs

Result: Radiologists and Fine-Tuned CNNs

Name	Acc	Sen	Spe
Radiologist 1 by TIRADS	87.3	95.0	72.0
Radiologist 2 by TIRADS	80.0	79.0	82.0
Radiologist 1 by Kim et al	84.7	82.0	90.0
Radiologist 2 by Kim et al	80.0	79.0	82.0
AlexNet	86.7	88.0	86.0
OverFeat	85.3	84.0	86.0
VGG	86.0	84.0	87.0
VGG-verydeep	85.3	74.0	91.0
ResNet	84.0	86.0	83.0
Inception	86.7	78.0	91.0

Reference: Eunjung Lee, Heonkyu Ha, Hye Jung Kim, Hee Jung Moon, Jung Hee Byon, Sun Huh, Jinwoo Son, Jiyoung Yoon, Kyunghwa Han, Jin Young Kwak, Differentiation of thyroid nodules on US using features learned and extracted from various convolutional neural networks, Scientific Reports, 9 (2019 December) 19854-1-19854-11

Result: Feature Extraction with and without Fine-Tuning

Net	Name	#	Without fine-tuning			With fine-tuning		
			Acc	Sen	Spe	Acc	Sen	Spe
AlexNet	fc1-SVM	4096	80.7	80.0	81.0	**87.3**	86.0	88.0
	fc1-RF	4096	85.3	82.0	87.0	86.0	82.0	88.0
	fc2-SVM	4096	81.3	80.0	82.0	84.7	82.0	86.0
	fc2-RF	4096	84.0	78.0	87.0	**87.3**	82.0	**90.0**
	fc1fc2-SVM	8192	82.0	80.0	83.0	85.3	84.0	86.0
	fc1fc2-RF	8192	86.0	82.0	88.0	**87.3**	84.0	89.0
	CNN					86.7	**88.0**	86.0
OverFeat	fc1-SVM	4096	78.7	74.0	81.0	84.7	82.0	86.0
	fc1-RF	4096	81.3	78.0	83.0	87.3	84.0	89.0
	fc2-SVM	4096	81.3	80.0	82.0	85.3	84.0	86.0
	fc2-RF	4096	81.3	74.0	85.0	**88.0**	84.0	**90.0**
	fc1fc2-SVM	8192	81.3	76.0	84.0	85.3	84.0	86.0
	fc1fc2-RF	8192	82.0	72.0	87.0	**88.0**	**86.0**	89.0
	CNN					85.3		
VGG	fc1-SVM	4096	79.3	82.0	78.0	84.7	80.0	87.0
	fc1-RF	4096	84.7	80.0	87.0	**89.3**	**86.0**	**91.0**
	fc2-SVM	4096	80.7	84.0	79.0	86.0	82.0	88.0
	fc2-RF	4096	85.3	80.0	88.0	88.0	84.0	90.0
	fc1fc2-SVM	8192	79.3	82.0	78.0	86.7	82.0	89.0
	fc1fc2-RF	8192	82.7	80.0	84.0	88.7	84.0	**91.0**
	CNN					86.0		

Net	Name	#	Without fine-tuning			With fine-tuning		
			Acc	Sen	Spe	Acc	Sen	Spe
VGG-verydeep	fc1-SVM	4096	84.7	88.0	83.0	78.0	76.0	79.0
	fc1-RF	4096	84.0	88.0	82.0	74.0	76.0	73.0
	fc2-SVM	4096	84.0	88.0	82.0	72.0	76.0	70.0
	fc2-RF	4096	**85.3**	90.0	83.0	69.3	74.0	67.0
	fc1fc2-SVM	8192	84.7	88.0	83.0	76.0	74.0	77.0
	fc1fc2-RF	8192	**85.3**	**92.0**	82.0	71.3	74.0	70.0
	CNN					**85.3**	74.0	**91.0**
ResNet	avg-SVM	2048	84.0	82.0	**85.0**	74.7	82.0	71.0
	avg-RF	2048	**85.3**	**86.0**	**85.0**	76.7	80.0	75.0
	CNN					84.0	**86.0**	83.0
Inception	avg-SVM	2048	85.3	**82.0**	87.0	74.7	62.0	81.0
	avg-RF	2048	84.7	72.0	**91.0**	76.0	68.0	80.0
	CNN					**86.7**	78.0	**91.0**

Result: Selected Features for Feature Concatenation and Classification Ensemble

- AlexNet
  fc2 with fine-tuning ([A])
- OverFeat
  fc2 with fine-tuning ([O])
- VGG
  fc1 with fine-tuning ([V])
- VGG-verydeep
  fc2 without fine-tuning ([Vv])
- ResNet
  avg without fine-tuning ([R])
- Inception
  avg without fine-tuning ([I])

Name	Classifier SVM Acc	SVM Sen	SVM Spe	RF Acc	RF Sen	RF Spe
[A]	84.7	82.0	86.0	87.3	82.0	90.0
[O]	85.3	84.0	86.0	88.0	84.0	90.0
[V]	84.7	80.0	87.0	89.3	86.0	91.0

Name	Classifier SVM Acc	SVM Sen	SVM Spe	RF Acc	RF Sen	RF Spe
[Vv]	84.0	88.0	82.0	85.3	90.0	83.0
[R]	84.0	82.0	85.0	85.3	86.0	85.0
[I]	85.3	82.0	87.0	84.7	72.0	91.0

Result: Feature Concatenation (2~3 CNNs)

Name	SVM Acc	SVM Sen	SVM Spe	RF Acc	RF Sen	RF Spe
[AO]	86.7	84.0	88.0	87.3	82.0	90.0
[AV]	88.0	76.0	94.0	90.7	90.0	91.0
[AVv]	93.3	90.0	95.0	92.7	90.0	94.0
[AR]	92.0	90.0	93.0	88.7	88.0	89.0
[AI]	88.0	86.0	89.0	90.7	82.0	95.0
[OV]	90.0	82.0	94.0	92.0	90.0	93.0
[OVv]	90.7	90.0	91.0	92.0	92.0	92.0
[OR]	92.0	90.0	93.0	89.3	88.0	90.0
[OI]	87.3	84.0	89.0	90.7	86.0	93.0
[VVv]	90.7	88.0	92.0	93.3	90.0	95.0
[VR]	92.0	86.0	95.0	90.0	86.0	92.0
[VI]	87.3	80.0	91.0	88.7	78.0	94.0
[VvR]	88.7	90.0	88.0	84.7	88.0	83.0
[VvI]	84.0	80.0	86.0	86.7	82.0	89.0
[RI]	85.3	82.0	87.0	85.3	78.0	89.0

Name	SVM Acc	SVM Sen	SVM Spe	RF Acc	RF Sen	RF Spe
[AOV]	90.7	82.0	95.0	91.3	92.0	91.0
[AOVv]	91.3	88.0	93.0	93.3	92.0	94.0
[AOR]	92.0	90.0	93.0	86.7	82.0	89.0
[AOI]	88.7	86.0	90.0	93.3	84.0	98.0
[AVVv]	93.3	88.0	96.0	93.3	90.0	95.0
[AVR]	93.3	84.0	98.0	92.0	88.0	94.0
[AVI]	92.0	86.0	95.0	92.7	80.0	**99.0**
[AVvR]	93.3	90.0	95.0	92.7	90.0	94.0
[AVvI]	90.0	88.0	91.0	91.3	86.0	94.0
[ARI]	88.7	86.0	90.0	90.7	84.0	94.0
[OVVv]	92.0	88.0	94.0	93.3	90.0	95.0
[OVR]	**94.7**	88.0	98.0	93.3	**94.0**	93.0
[OVI]	90.7	82.0	95.0	91.3	82.0	96.0
[OVvR]	90.7	90.0	91.0	91.3	90.0	92.0
[OVvI]	88.7	84.0	91.0	91.3	86.0	94.0
[ORI]	88.0	84.0	90.0	88.7	80.0	93.0
[VVvR]	92.0	88.0	94.0	93.3	90.0	95.0
[VVvI]	90.0	84.0	93.0	92.7	86.0	96.0
[VRI]	88.0	82.0	91.0	90.0	78.0	96.0
[VvRI]	86.7	84.0	88.0	87.3	82.0	90.0

Result: Feature Concatenation (4~6 CNNs)

Name	SVM Acc	SVM Sen	SVM Spe	RF Acc	RF Sen	RF Spe
[AOVVv]	93.3	88.0	96.0	**94.0**	92.0	95.0
[AOVR]	93.3	84.0	98.0	92.7	92.0	93.0
[AOVI]	92.7	84.0	97.0	92.7	84.0	97.0
[AOVvR]	92.0	90.0	93.0	92.7	90.0	94.0
[AOVvI]	91.3	86.0	94.0	91.3	86.0	94.0
[AORI]	89.3	86.0	91.0	91.3	<u>84.0</u>	95.0
[AVVvR]	**94.0**	90.0	96.0	92.7	<u>88.0</u>	95.0
[AVVvI]	91.3	88.0	93.0	92.0	<u>88.0</u>	94.0
[AVRI]	93.3	86.0	97.0	92.0	<u>84.0</u>	96.0
[AVvRI]	90.0	88.0	91.0	92.0	<u>86.0</u>	95.0
[OVVvR]	93.3	90.0	95.0	**94.0**	**94.0**	94.0
[OVVvI]	90.0	<u>84.0</u>	93.0	91.3	<u>86.0</u>	94.0
[OVRI]	92.0	84.0	96.0	92.0	<u>78.0</u>	**99.0**
[OVvRI]	90.0	88.0	91.0	92.0	<u>88.0</u>	94.0
[VVvRI]	90.0	<u>84.0</u>	93.0	90.0	<u>84.0</u>	93.0

Name	SVM Acc	SVM Sen	SVM Spe	RF Acc	RF Sen	RF Spe
[AOVVvR]	**94.0**	90.0	96.0	93.3	90.0	95.0
[AOVVvI]	93.3	90.0	95.0	92.7	<u>88.0</u>	95.0
[AOVRI]	93.3	86.0	97.0	93.3	86.0	97.0
[AOVvRI]	92.0	88.0	94.0	92.7	<u>88.0</u>	95.0
[AVVvRI]	91.3	88.0	93.0	92.0	<u>88.0</u>	94.0
[OVVvRI]	92.0	88.0	94.0	90.7	<u>86.0</u>	93.0
[AOVVvRI]	93.3	90.0	95.0	90.7	<u>86.0</u>	93.0

Result: Classification Ensemble

Name	Acc	Sen	Spe	Name	Acc	Sen	Spe	Name	Acc	Sen	Spe
$[A]$	86.0	82.0	88.0	$[A][O][V]$	90.7	90.0	91.0	$[A][O][V][Vv]$	93.3	90.0	95.0
$[O]$	86.0	84.0	87.0	$[A][O][Vv]$	91.3	90.0	92.0	$[A][O][V][R]$	93.3	88.0	96.0
$[V]$	87.3	82.0	90.0	$[A][O][R]$	89.3	90.0	89.0	$[A][O][V][I]$	93.3	88.0	96.0
$[Vv]$	83.3	88.0	81.0	$[A][O][I]$	89.3	86.0	91.0	$[A][O][Vv][R]$	92.0	88.0	94.0
$[R]$	84.7	90.0	82.0	$[A][V][Vv]$	92.0	86.0	95.0	$[A][O][Vv][I]$	93.3	88.0	96.0
$[I]$	84.0	70.0	91.0	$[A][V][R]$	**94.7**	90.0	97.0	$[A][O][R][I]$	92.0	86.0	95.0
$[A][O]$	84.7	80.0	87.0	$[A][V][I]$	92.7	84.0	97.0	$[A][V][Vv][R]$	93.3	88.0	96.0
$[A][V]$	90.7	80.0	96.0	$[A][Vv][R]$	90.7	90.0	91.0	$[A][V][Vv][I]$	92.0	84.0	96.0
$[A][Vv]$	92.0	88.0	94.0	$[A][Vv][I]$	90.7	86.0	93.0	$[A][V][R][I]$	94.0	88.0	97.0
$[A][R]$	93.3	88.0	96.0	$[A][R][I]$	92.7	86.0	96.0	$[A][Vv][R][I]$	90.0	88.0	91.0
$[A][I]$	90.7	82.0	95.0	$[O][V][Vv]$	93.3	88.0	96.0	$[O][V][Vv][R]$	92.0	90.0	93.0
$[O][V]$	91.3	86.0	94.0	$[O][V][R]$	92.7	88.0	95.0	$[O][V][Vv][I]$	92.7	86.0	96.0
$[O][Vv]$	90.7	88.0	92.0	$[O][V][I]$	94.0	86.0	**98.0**	$[O][V][R][I]$	92.0	86.0	95.0
$[O][R]$	90.7	88.0	92.0	$[O][Vv][R]$	90.0	90.0	90.0	$[O][Vv][R][I]$	89.3	88.0	90.0
$[O][I]$	92.7	84.0	97.0	$[O][Vv][I]$	89.3	86.0	91.0	$[V][Vv][R][I]$	88.7	86.0	90.0
$[V][Vv]$	90.0	88.0	91.0	$[O][R][I]$	91.3	86.0	94.0	$[A][O][V][Vv][R]$	94.0	90.0	96.0
$[V][R]$	92.0	86.0	95.0	$[V][Vv][R]$	89.3	88.0	90.0	$[A][O][V][Vv][I]$	94.0	90.0	96.0
$[V][I]$	89.3	76.0	96.0	$[V][Vv][I]$	88.7	82.0	92.0	$[A][O][V][R][I]$	94.0	88.0	97.0
$[Vv][R]$	88.7	**92.0**	87.0	$[V][R][I]$	92.0	82.0	97.0	$[A][O][Vv][R][I]$	91.3	88.0	93.0
$[Vv][I]$	86.7	86.0	87.0	$[Vv][R][I]$	87.3	86.0	88.0	$[A][V][Vv][R][I]$	93.3	88.0	96.0
$[R][I]$	86.0	84.0	87.0					$[O][V][Vv][R][I]$	90.7	88.0	92.0
								$[A][O][V][Vv][R][I]$	92.0	86.0	95.0

SERA

– Severance Diagnostic Helper based on Deeplearning designed by Yonsei-CSE with Matlab

Pre-trained nets in matlab

- alexnet, vgg16, vgg19, squeezenet, googlenet, inceptionv3, densenet201, mobilenetv2, resnet18, resnet50, resnet101, xception, inceptionresnetv2, shufflenet

Transfer learning by replacing final layers – Severance data are used

Training: 2004년부터 2019년까지 세브란스병원에서 수집한 **13,560**장의 갑상선 ROI 영상
7160(악성 결절)+7160(6400 양성 결절 + 760 random 좌우반전)

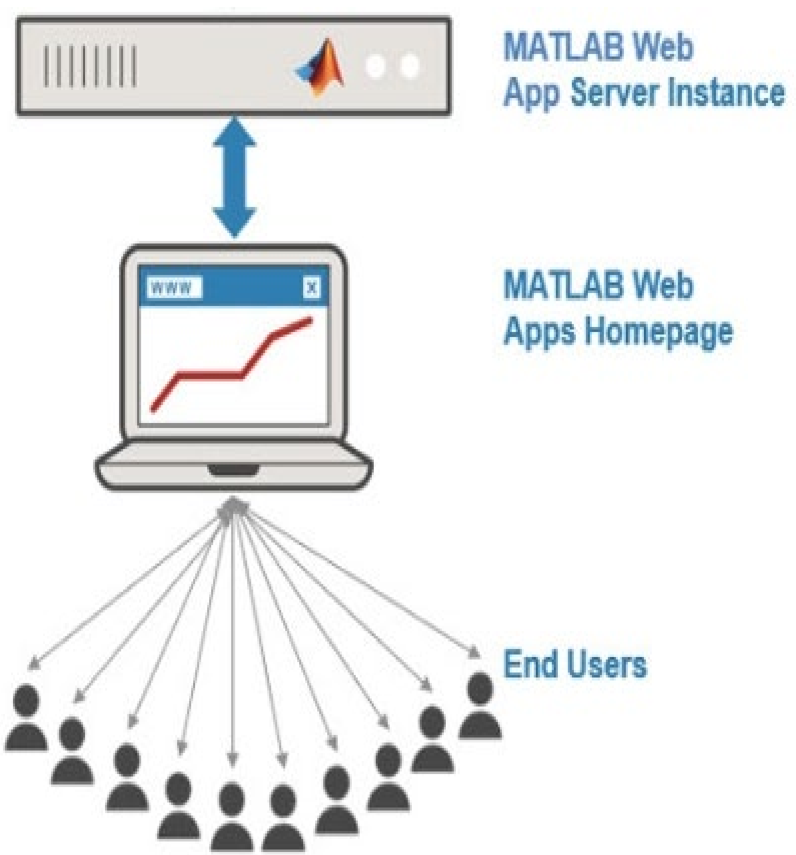
Test – Multicenter study
Internal test: 634 세브란스 test set
External test: 781 삼성의료원, 200 분당차병원, 200 경희대병원

Two individual CNNs and two classification ensembles (CNNE1 and CNNE2) were tested to differentiate malignant and benign thyroid nodules. CNNs demonstrated high area under the curves (AUCs) to diagnose malignant thyroid nodules (0.898-0.937 for the internal test set and 0.821-0.885 for the external test sets).
* 모든 환자데이터는 각 의료기관의 IRB 승인을 받아 진행하였습니다.

Reference: Jieun Koh, Eunjung Lee, Kyunghwa Han, Eun-Kyung Kim, Eun Ju Son, Yu-Mee Sohn, Mirinae Seo, Mi-ri Kwon, Jung Hyun Yoon, Jin Hwa Lee, Young Mi Park, Sungwon Kim, Jung Hee Shin, Jin Young Kwak, Diagnosis of thyroid nodules on ultrasonography by a deep convolutional neural network, Scientific Reports, 10(1) (2020 Sep) 15245

SERA

– Severance Diagnostic Helper based on Deeplearning designed by Yonsei-CSE with Matlab



- We developed software to determine the characteristics of thyroid tumors using MATLAB program.
- Using the MATLAB Web App server, we can deploy this software and users can use the program through the cell.
- This software has been made available through computer and mobile phone.

최종적으로 VGG16 net를 이용한 transfer learning과 training은 Matlab 2020a 버전 이용,

의학서버의 데이터 전송 및 갑상선 결절 악성률 예측값 전송은 MatlabWebAppServer(2020a),

웹 디자인 및 구성은 JavaScript기반을 통해 제작

SERA

– Severance Diagnostic Helper based on Deeplearning designed by Yonsei-CSE with Matlab

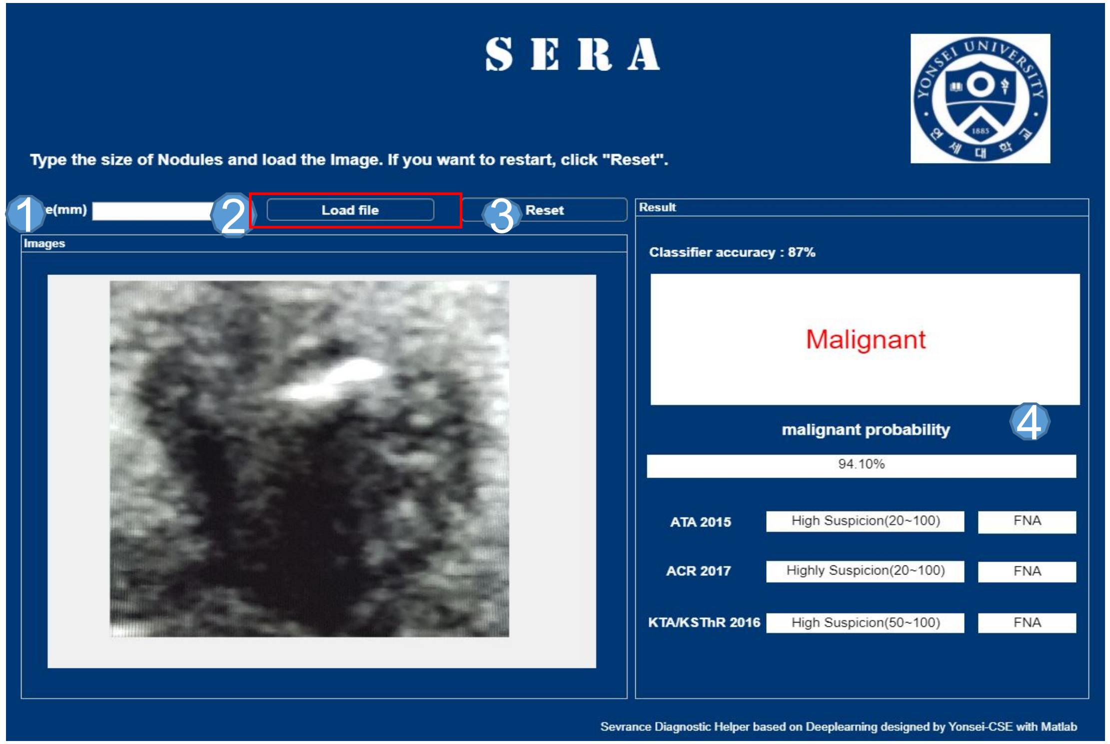


fig1) computer application

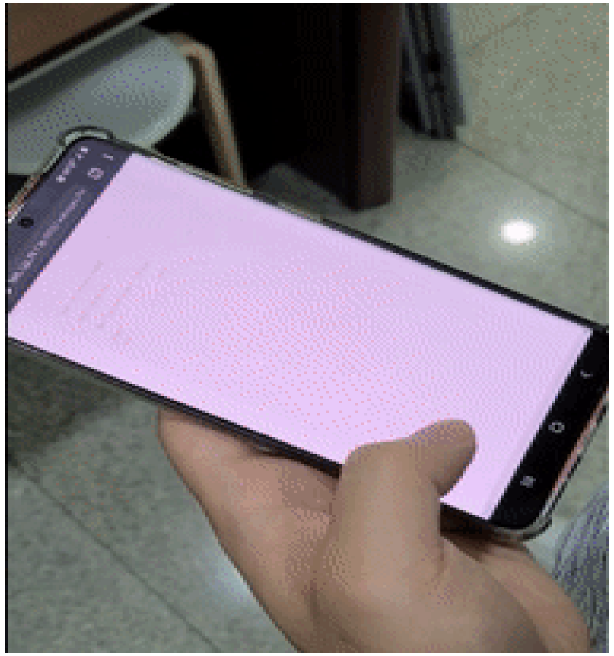


fig2) mobile application

1. In put the node size
2. click and upload the image
3. If you want to reset, click
4. you can see the results of the tumor and whether you need to FNA

Conclusion

For comparison diagnostic performance of experienced radiologists and CNNs, the output from CNNs is classified binary by using a Youden index with cutoff value.

Our results showed only fair interobserver agreement between the two experienced radiologists. These results may be because US is inherently dependent on the expertise of the performer, and because radiologists do not effectively use all available US findings to distinguish benign and malignant thyroid nodules.

To overcome these limitations of thyroid US, many automated US image analysis techniques have been extensively studied with different input data format, features, methods and classifiers. Recently, CNN, CNN combined with other classifiers, and other classifiers alone have been investigated for diagnosing

Therefore, this study demonstrates the high possibility of the CNN as a diagnostic or assistant tool for radiologists.

Future study includes various types of thyroid malignancies are needed with more data.

In conclusion, the CNNs showed comparable diagnostic performance compared to experienced radiologists in distinguishing malignant thyroid nodules from benign nodules on US. Therefore, the CNN can be employed as a useful method for the diagnosis of thyroid malignancy.