Görsel Tanımada Derin Öğrenme

Ceyhun Burak Akgül

Vispera Bilgi Teknolojileri & Boğaziçi Üniversitesi

www.vispera.co

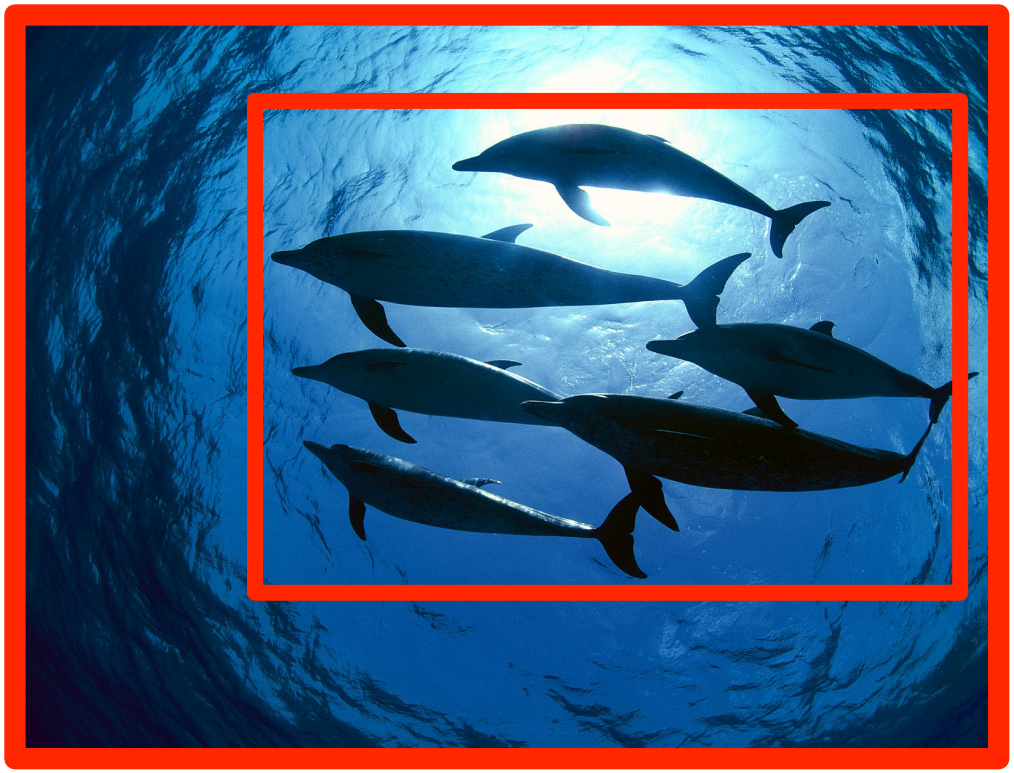
www.cba-research.com

Neyi Tanıyacağız?

Neyi Tanıyacağız?

Bir sualtı sahnesi

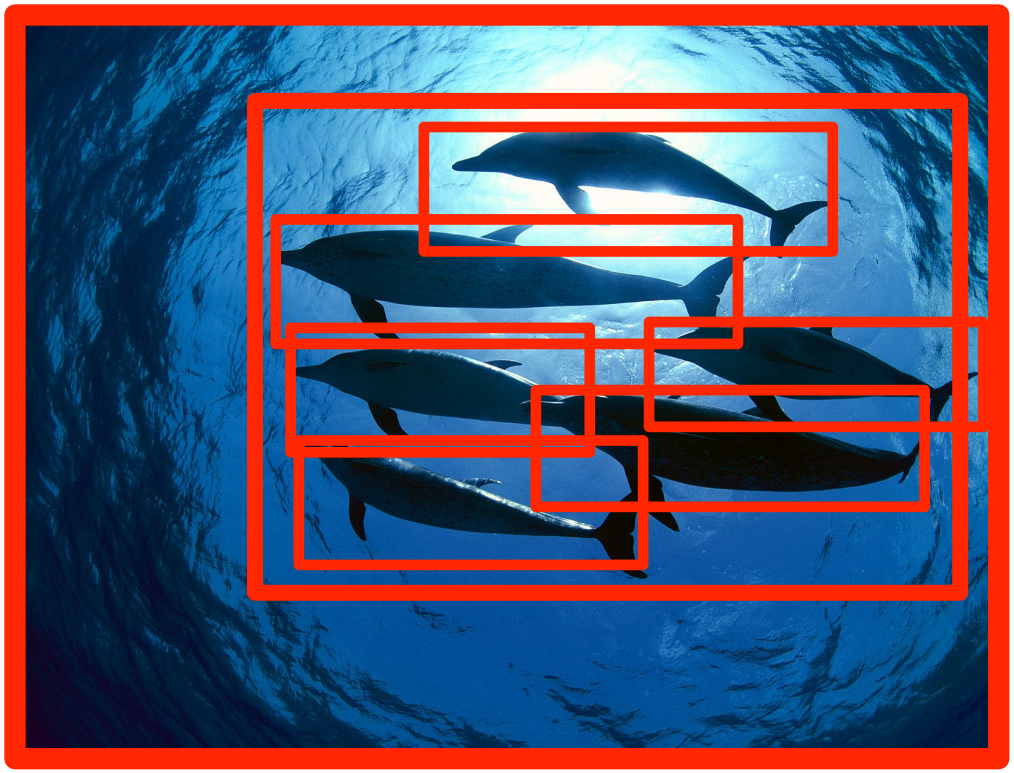
Neyi Tanıyacağız?



Bir sualtı sahnesi

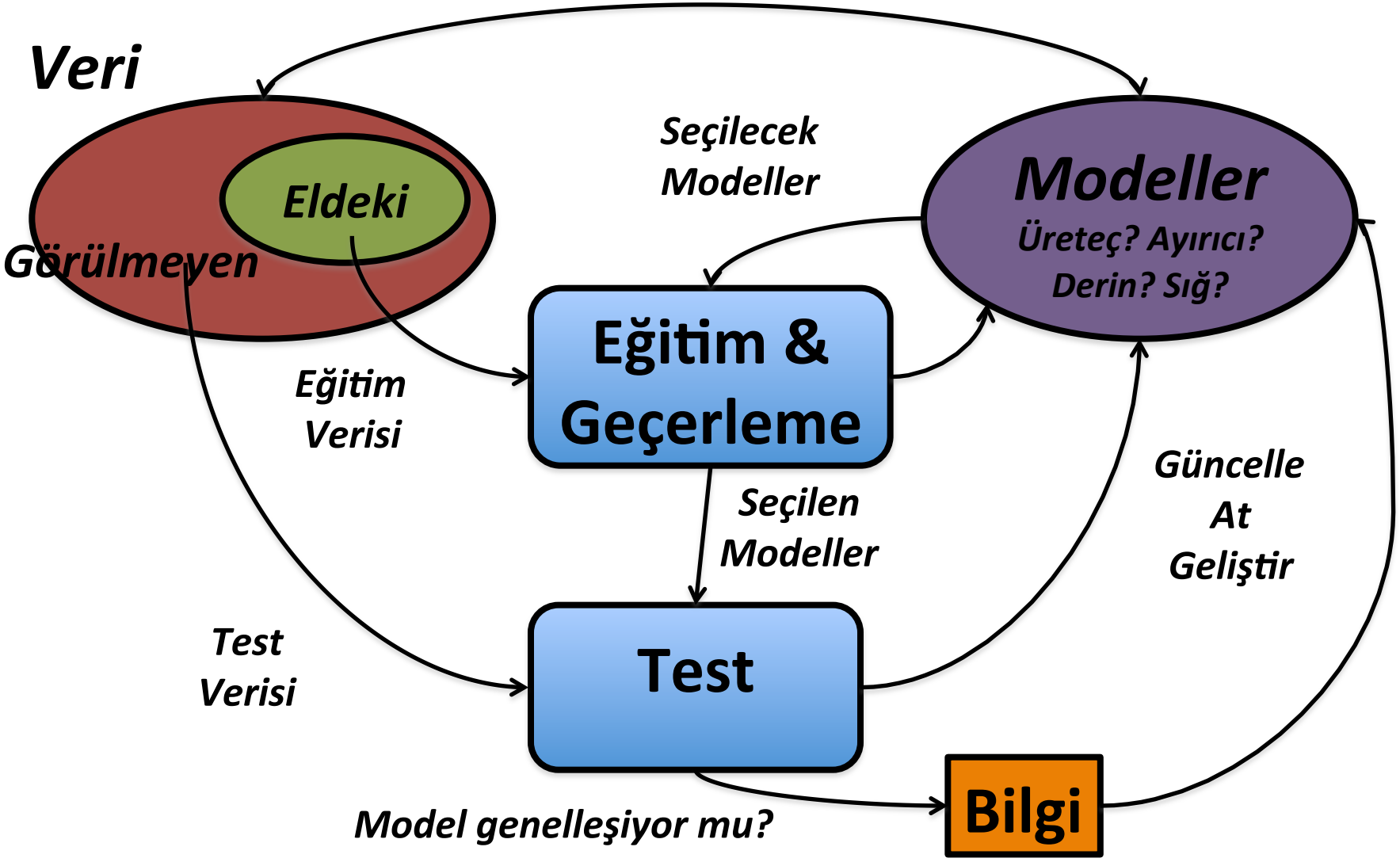
Yunus sürüsü

Neyi Tanıyacağız?



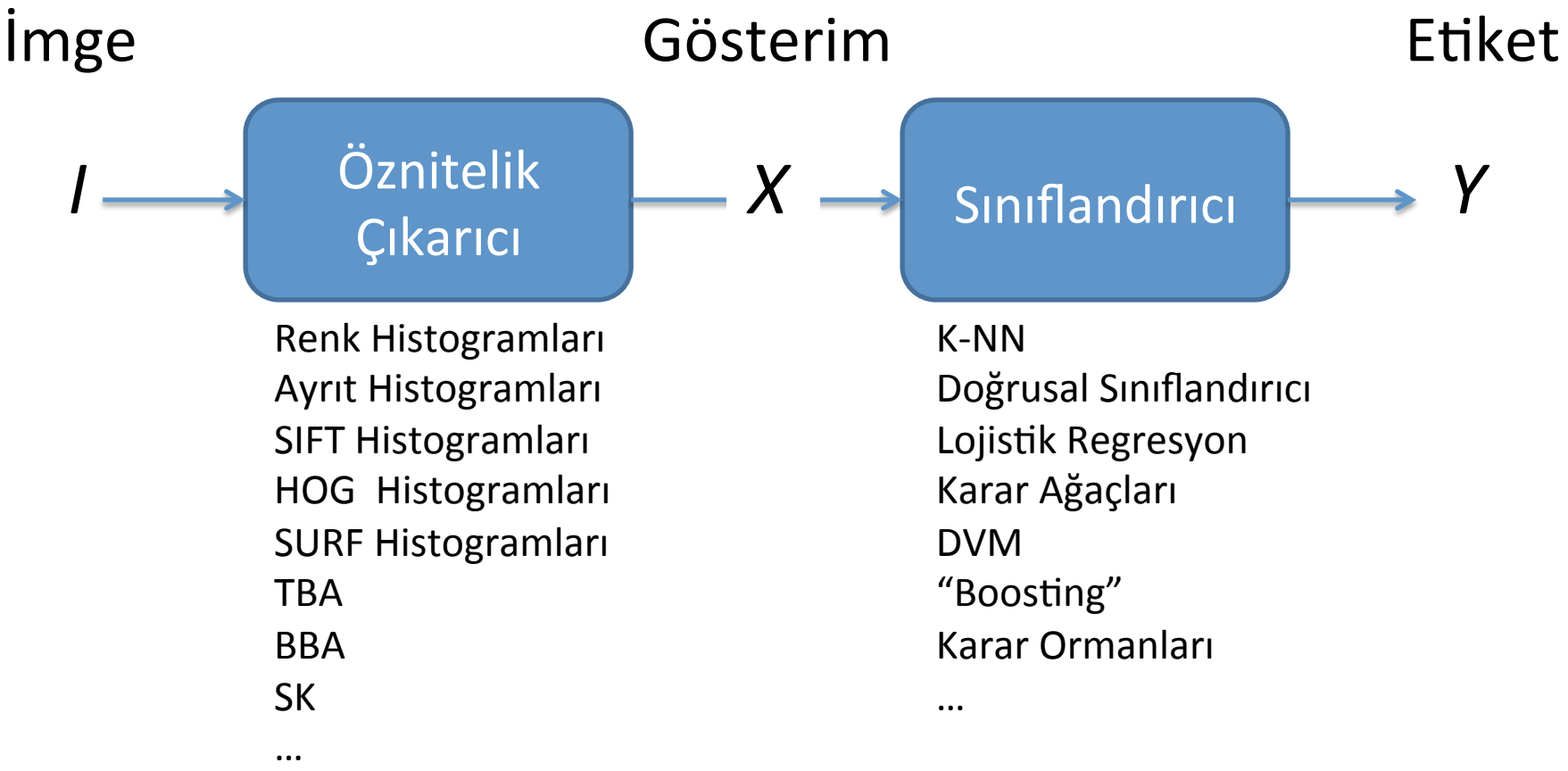
Bir sualtı sahnesi
Yunus sürüsü
6 adet yunus

Nasıl Tanıyacağız?

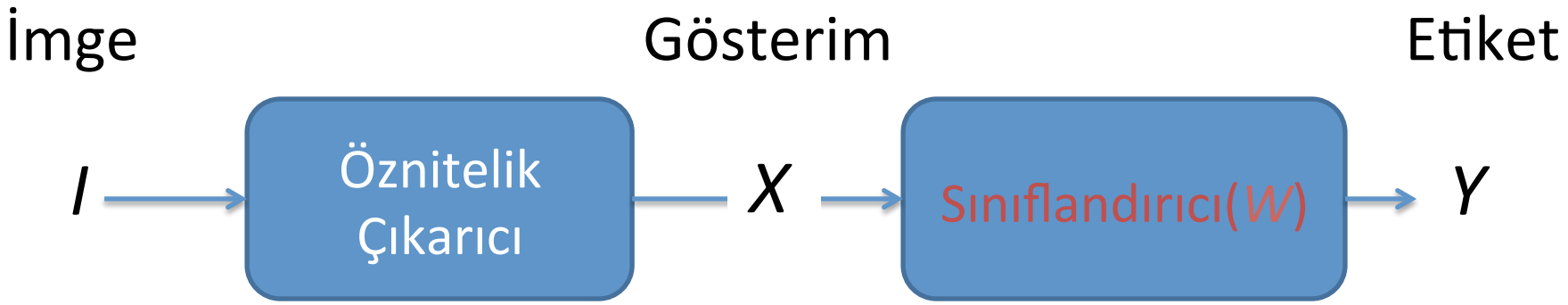


YAPAY ÖĞRENME SORUNSALI

Standart Tanıma Blokları ve Akışı



Sınıflandırıcıyı Öğrenmek



Sınıflandırıcı parametreleri *W* aşağıdaki ifadeyi enküçükleyerek öğrenilir

$$kayıp(Y, Y_{doğru}) + \lambda\ yapı(W)$$

Örnekler

$|Y\text{-}Y_{true}|_2$

$|Y\text{-}Y_{true}|_1$

$\max(0,1\text{-}Y\ Y_{true})$

$1\text{-}\mathbf{1}\{Y = Y_{true}\}$

...

Örnekler

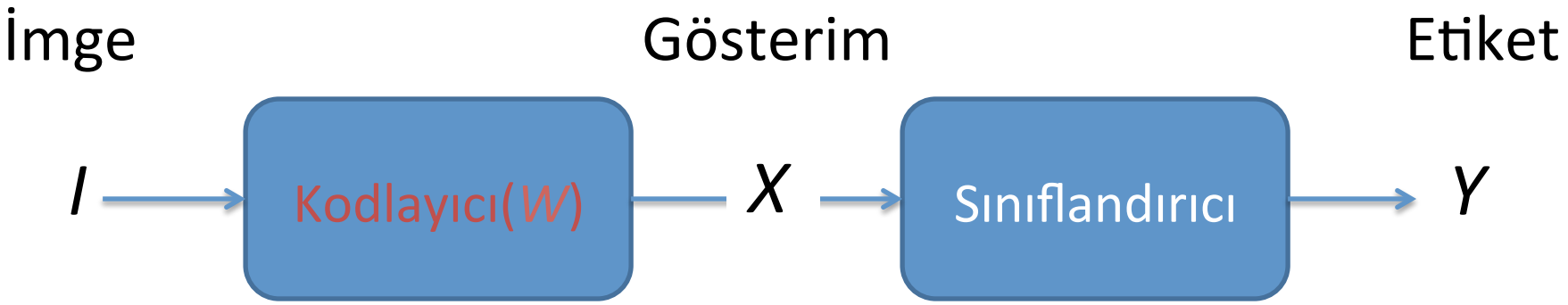
$|W|_2 = 1$

$|W|_1 < \rho$

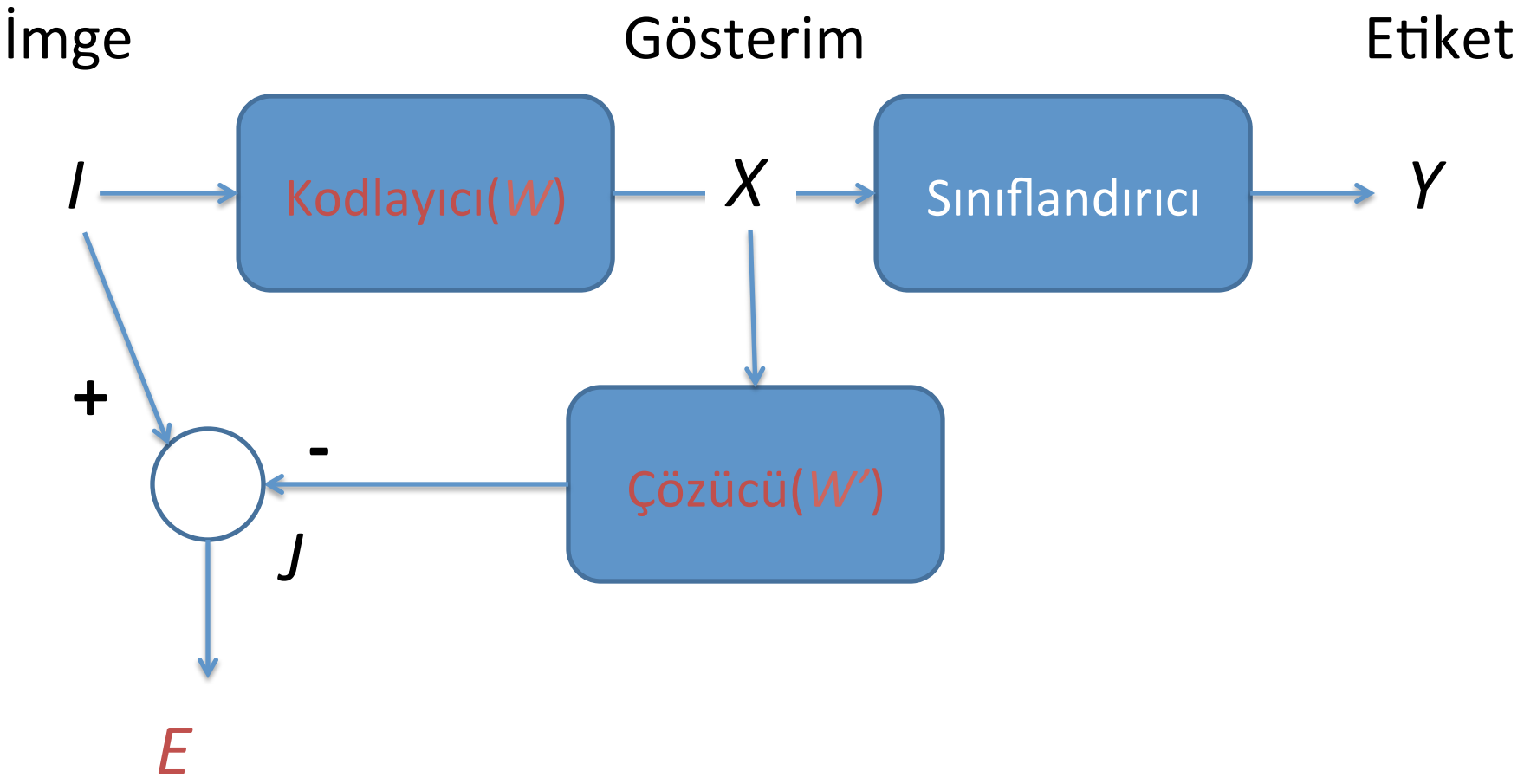
$|W|_o < \rho$

...

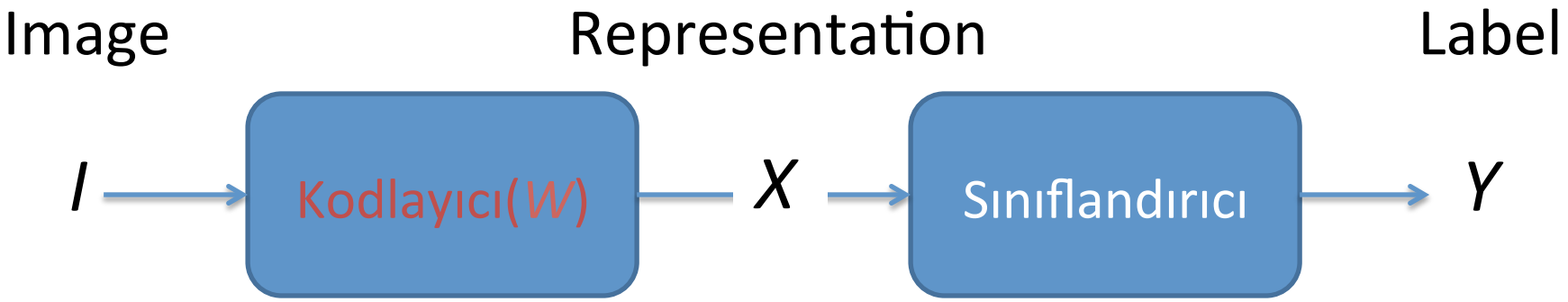
Gösterimi Öğrenmek – 1/6



Gösterimi Öğrenmek – 2/6



Gösterimi Öğrenmek – 3/6



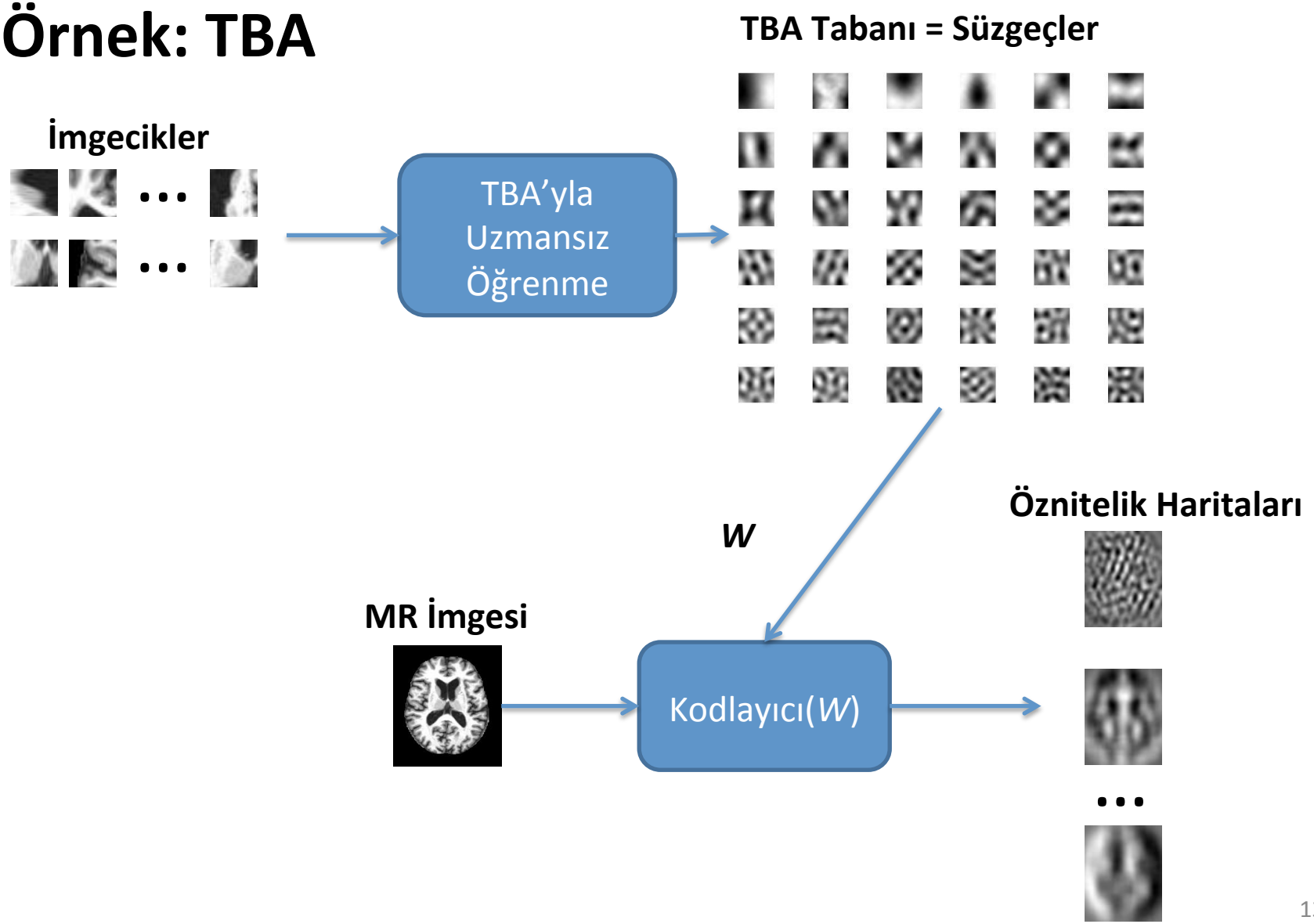
Gösterim *X* aşağıdaki ifadeyi enküçükleyerek öğrenilir

$$|E| + \lambda\ yapı(X,\ W)$$

Temel Bileşenler Analizi (TBA)
Bağımsız Bileşenler Analizi (BBA)
Seyrek Kodlama
...

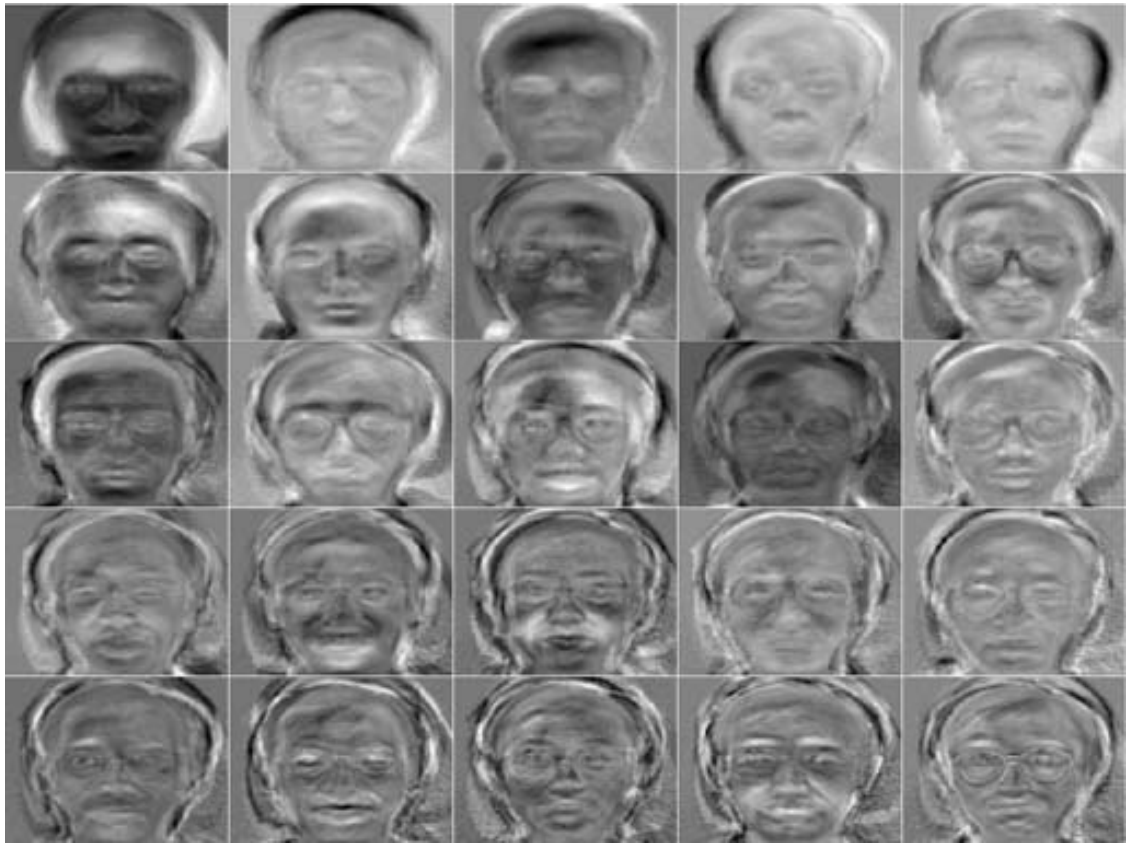
Gösterimi Öğrenmek – 4/6

Örnek: TBA



Gösterimi Öğrenmek – 5/6

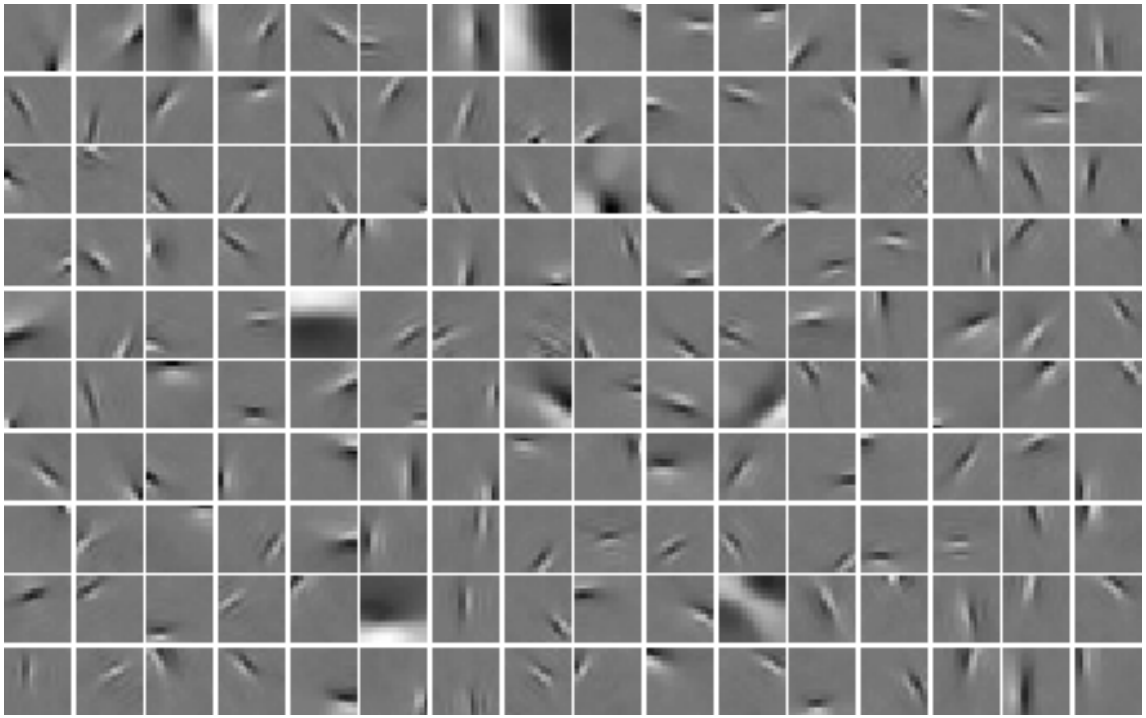
Örnek: Öz-Yüzler (Eigen-Faces)



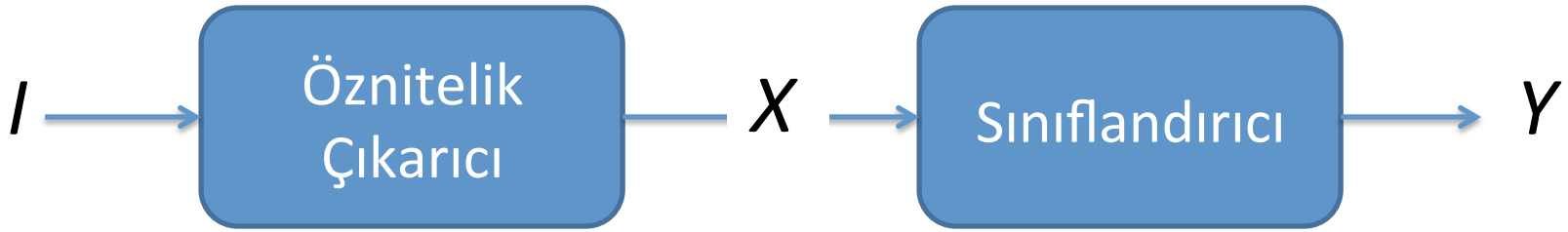
Gösterimi Öğrenmek – 6/6

Örnek: BBA (Bağımsız Bileşenler Analizi)

BBA Tabanı = Süzgeçler



Baştan Sona Öğrenmek

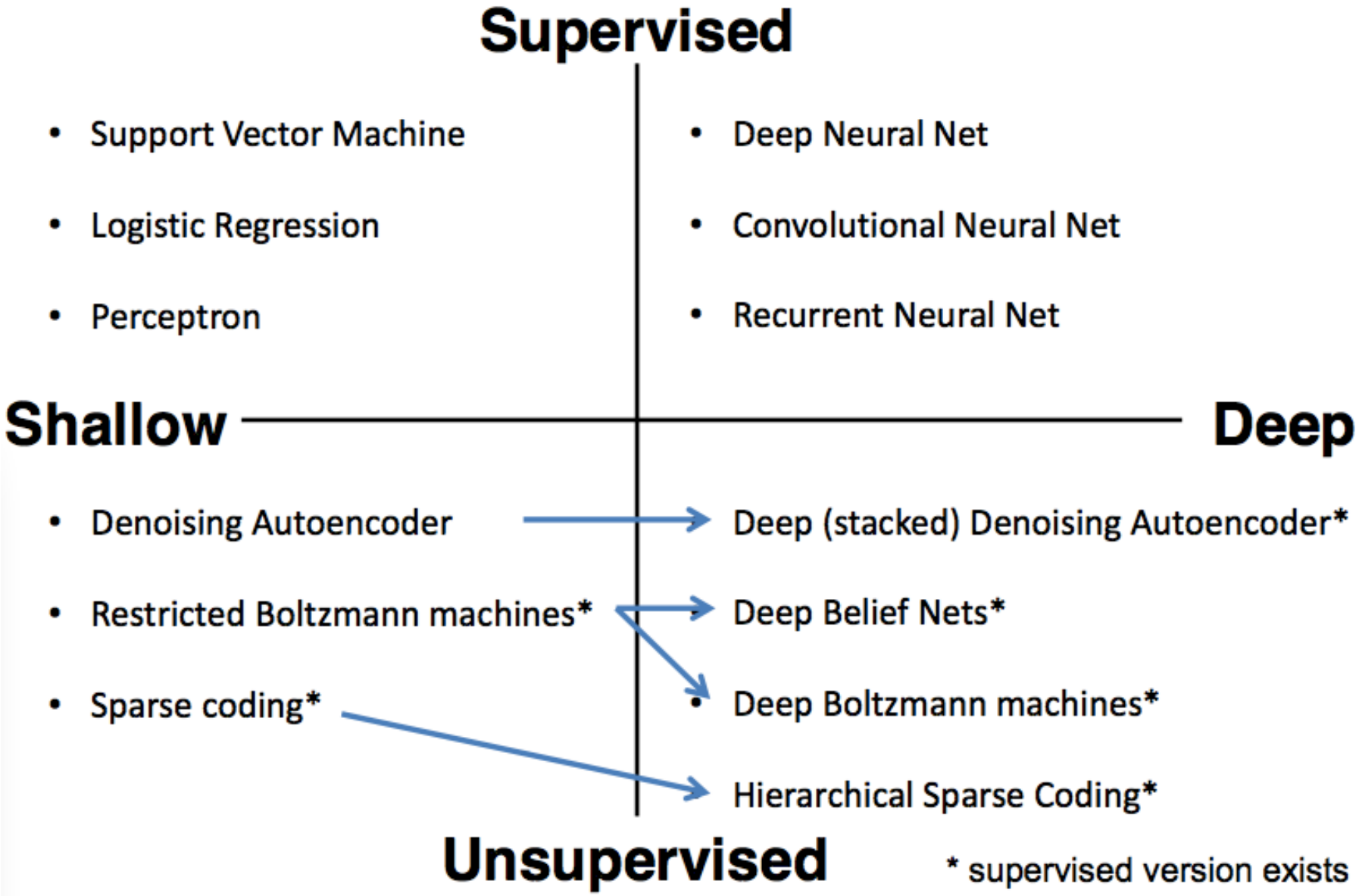


SIĞ

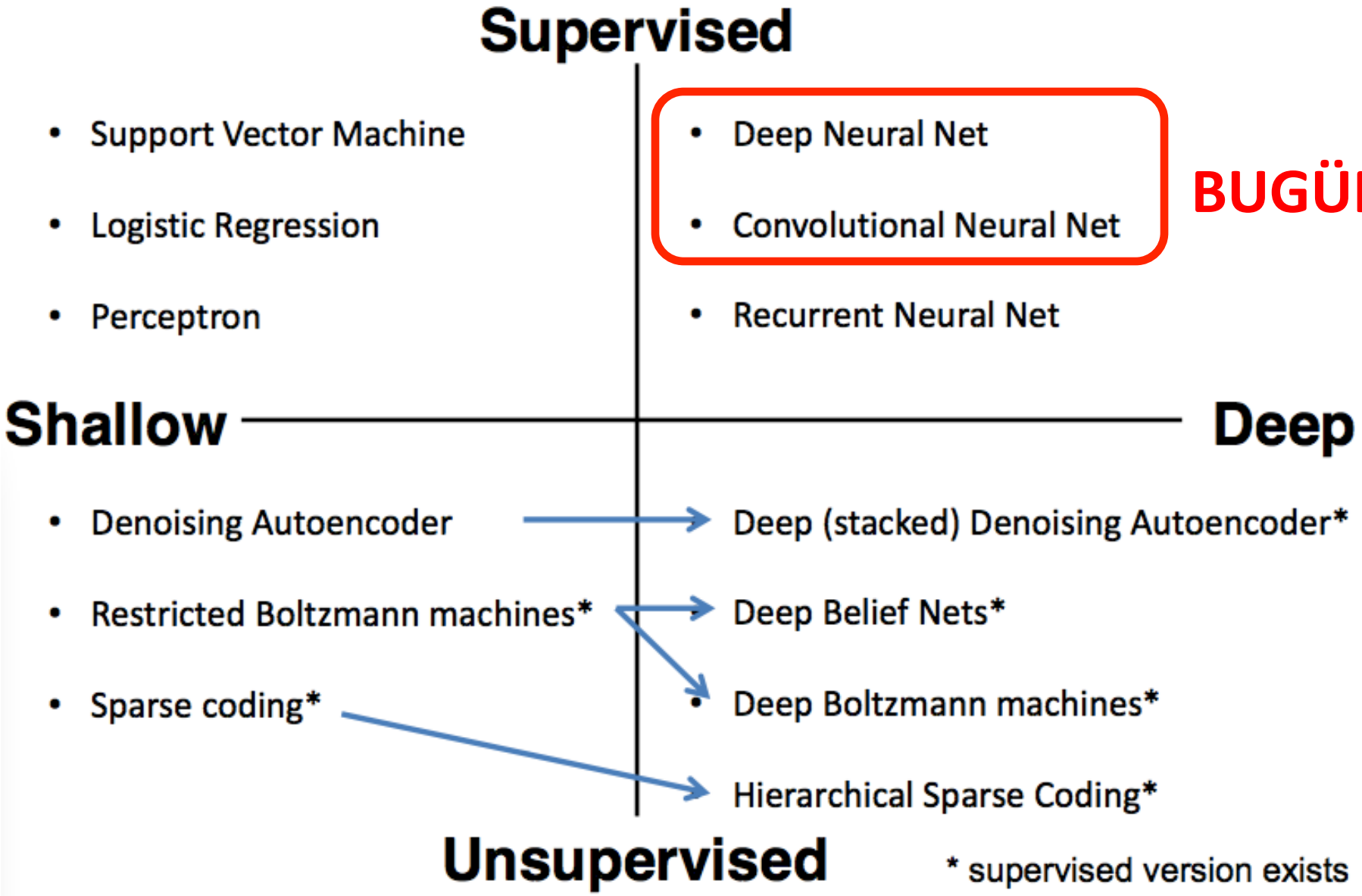
DERİN

- Derin Modeller: Sıradüzensel öznitelik çıkarıcılar
- Her katman bir öncekinin çıktısını girdi olarak kullanır
- Öğrenme baştan sona piksellerden etiketlere gerçekleşir

Öğrenme Yöntemleri Haritası

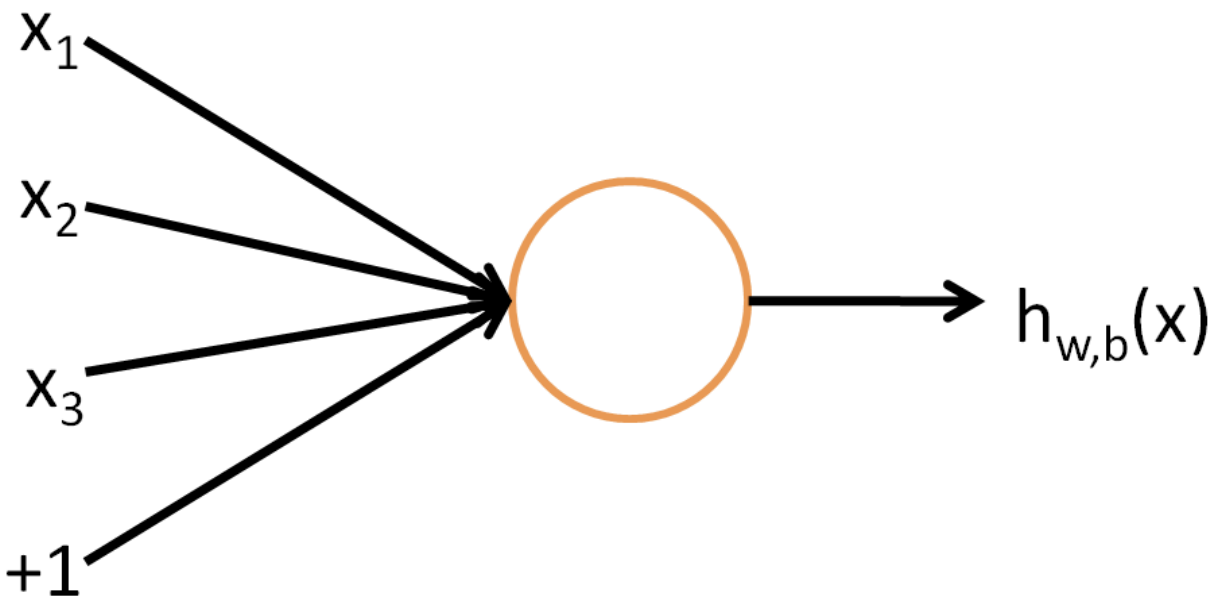


Öğrenme Yöntemleri Haritası



DERİN SİNİR AĞLARI

Perseptron



$$h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b)$$

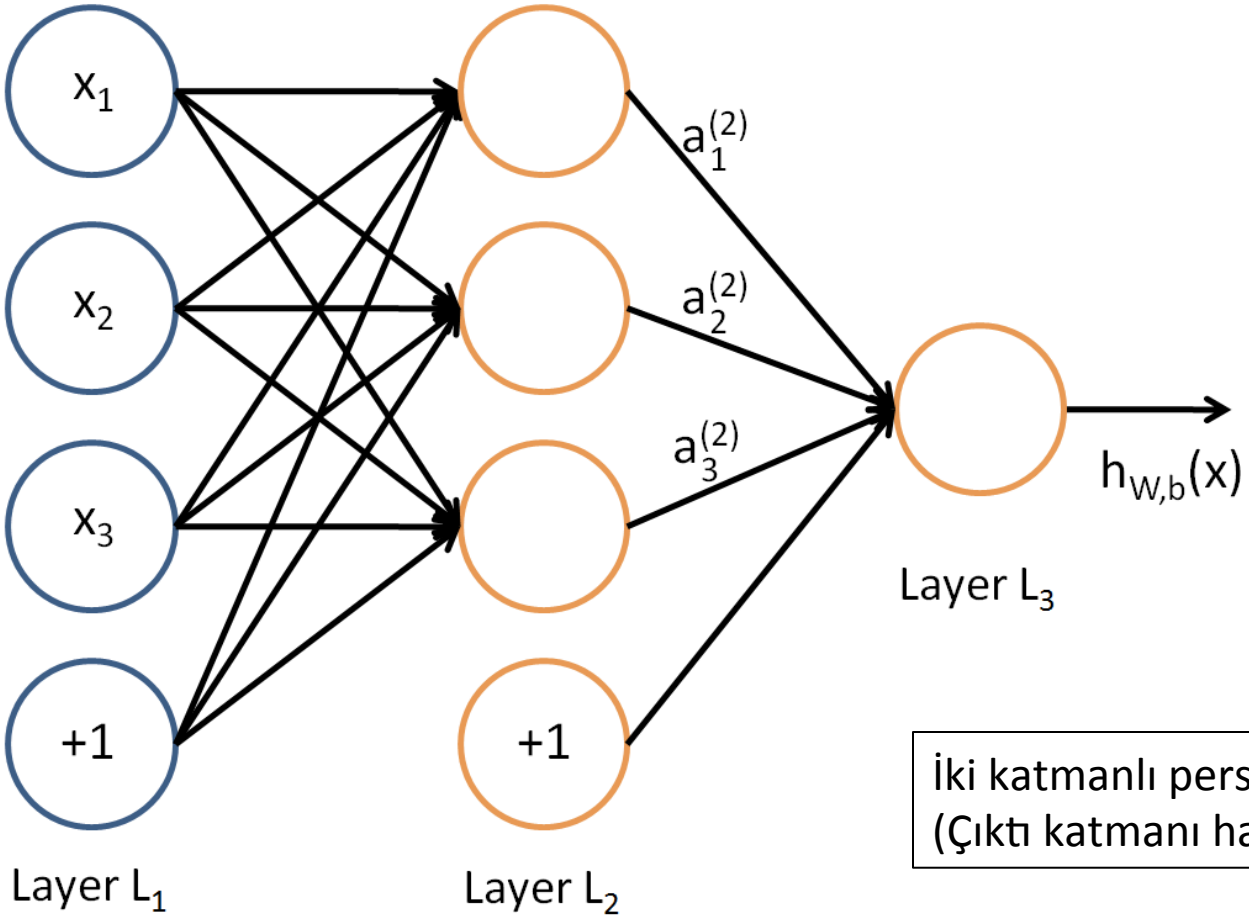
$$f(z) = \frac{1}{1 + \exp(-z)}$$

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$f(z) = \max(0, x)$$

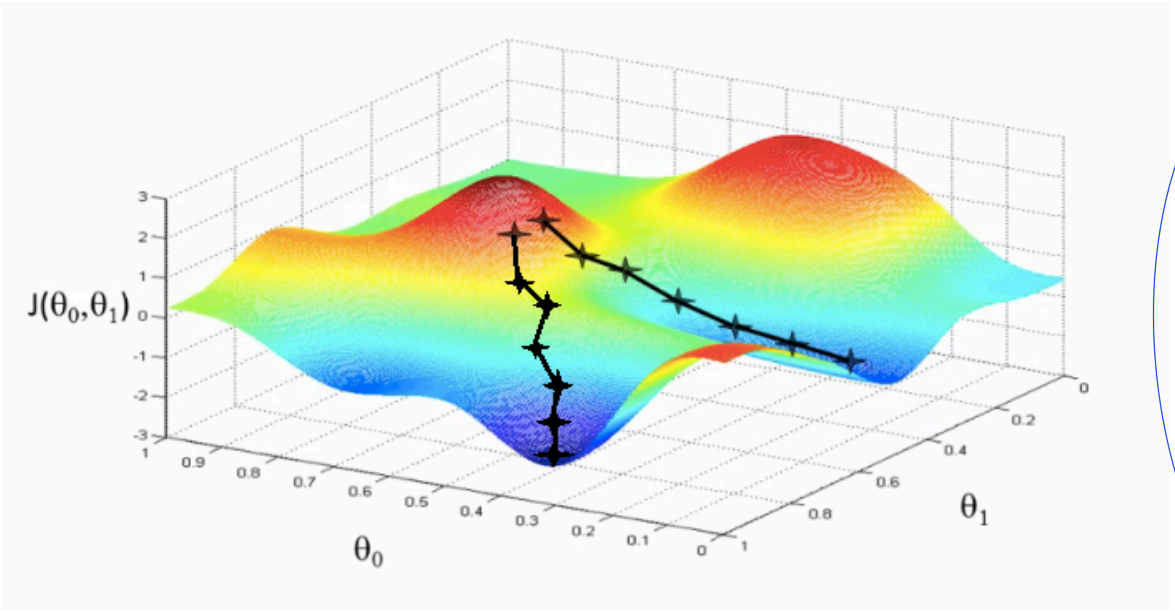
Derin Sinir Ağı

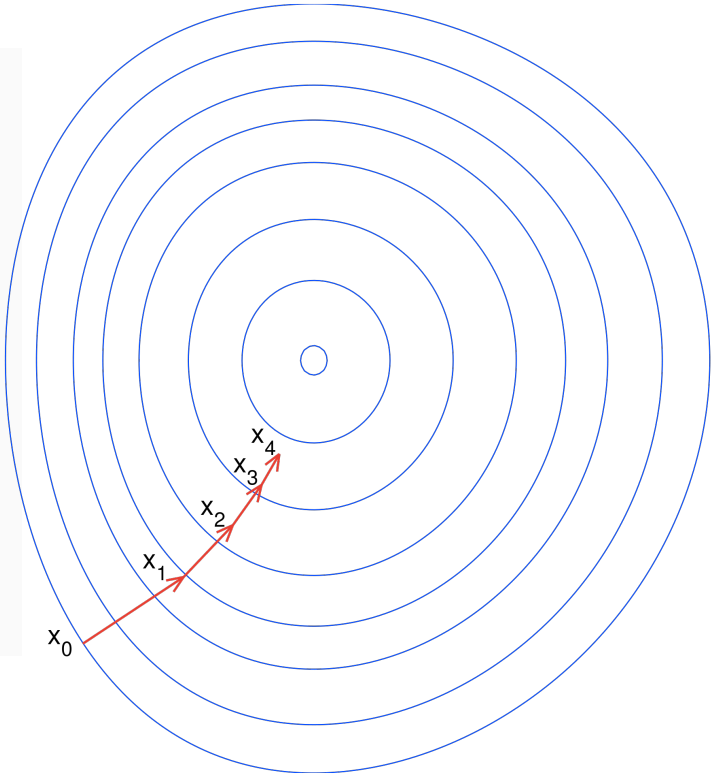
Derin Sinir Ağı = Çok Katmanlı Perseptron (ÇKP)



İki katmanlı perseptron
(Çıktı katmanı hariç)

Gradyan İnişi





$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W,b)$$

$$b_i^{(l)} = b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(W,b)$$

ÇKP'yi Eğitmek

"**Backpropagation**" Algoritması

Given a labeled training set and a criterion J

Randomly initialize the network parameters

Iterate until convergence

(1) **Forward pass:** Calculate the net output with the current set of parameters

(2) **Back propagation:** Calculate the error signals, hence the derivatives wrt the criterion *J*

(3) **Update:** Apply the update on the current set of parameters to obtain new parameters

ÇKP'yi Eğitmek

"**Backpropagation**" Algoritması

Given a labeled training set and a criterion J

Randomly initialize the network parameters

Iterate until convergence

(1) **Forward pass:** Calculate the net output with the current set of parameters

(2) **Back propagation:** Calculate the error signals, hence the derivatives wrt the criterion *J*

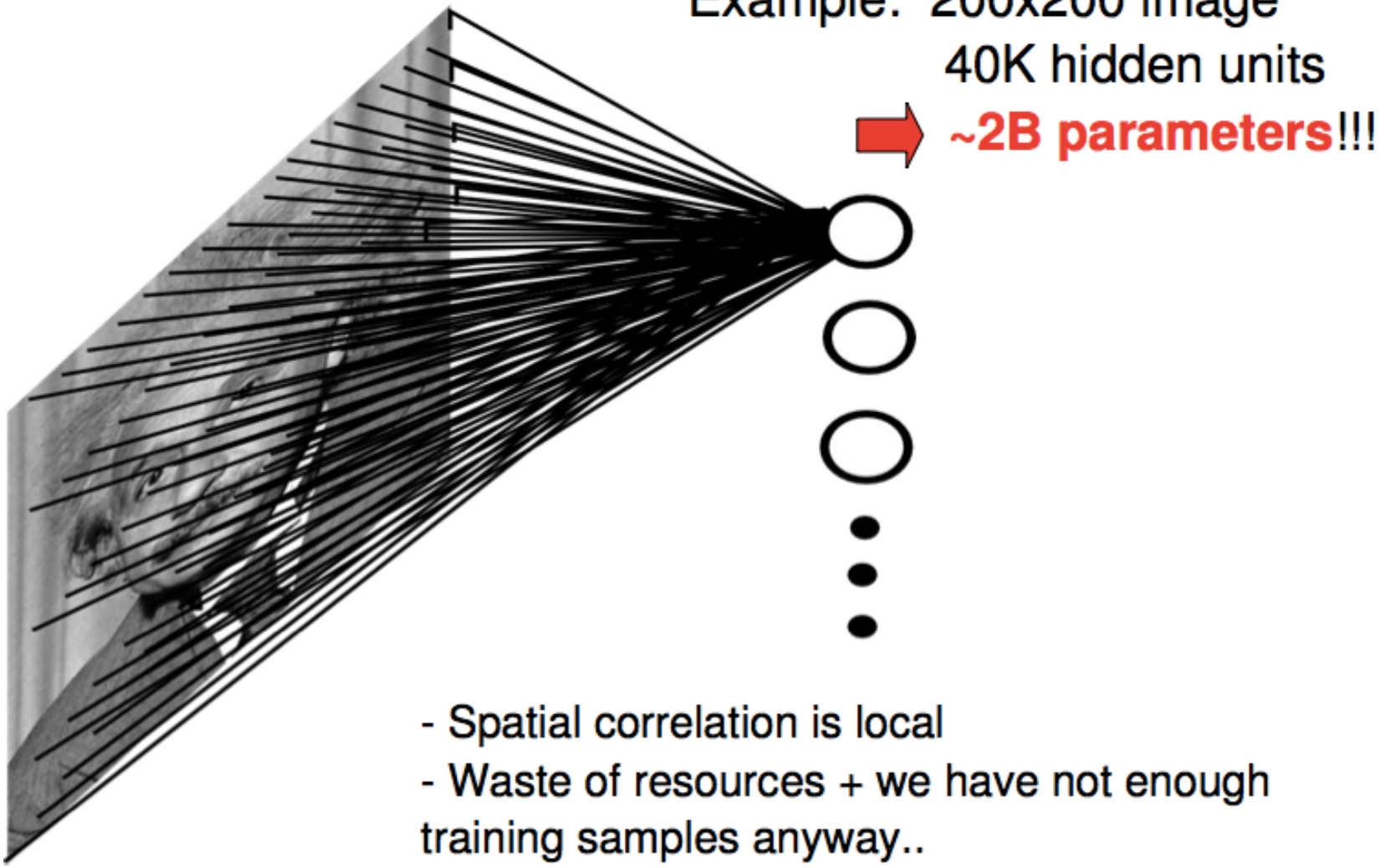
(3) **Update:** Apply the update on the current set of parameters to obtain new parameters

Yakın geçmişe kadar büyük ağlarda pek iyi çalışmadı

EVRİŞİMSEL SİNİR AĞLARI

Bağlantı Yapıları – 1/4

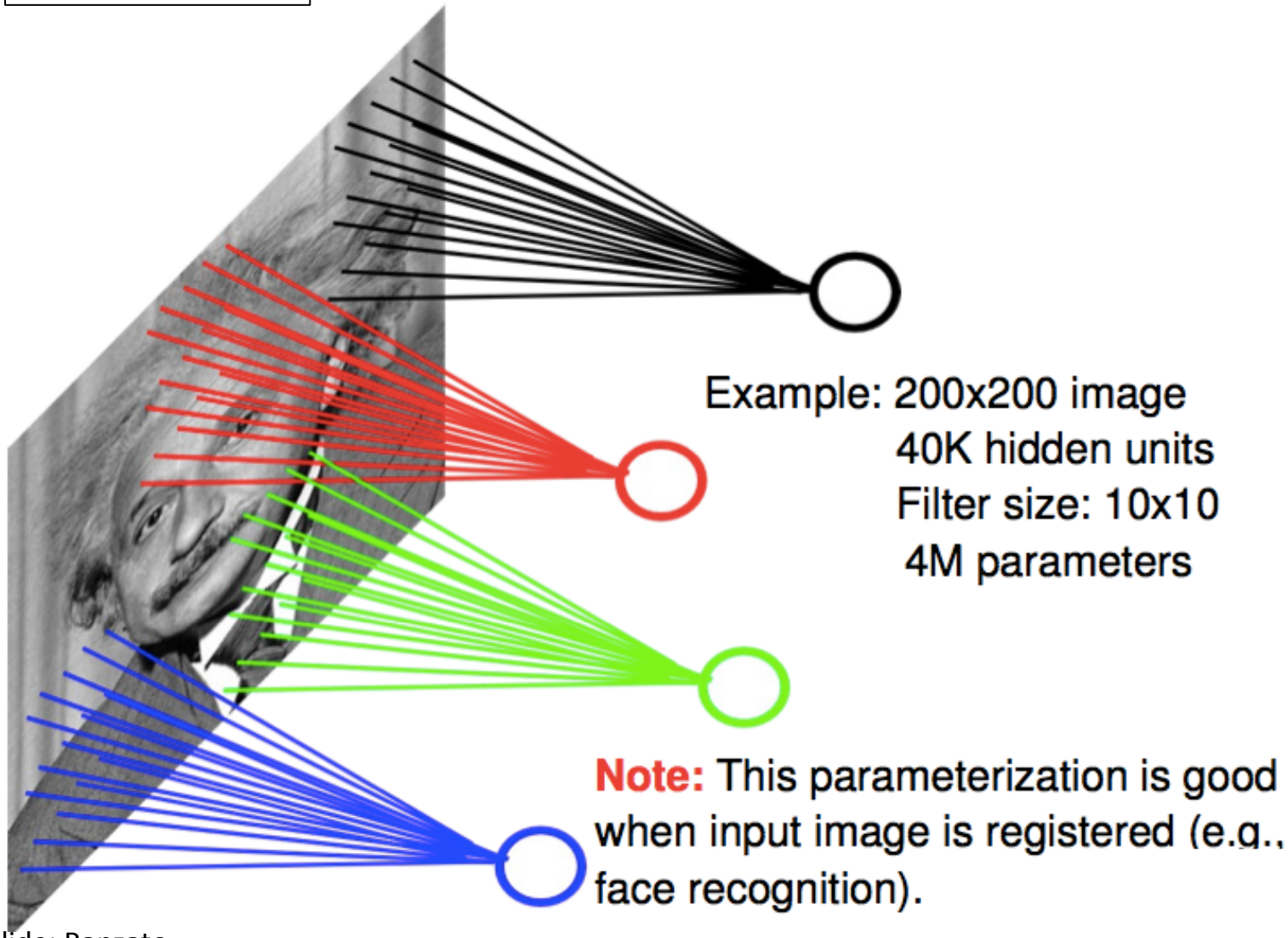
Tam Bağlı



Slide: Ranzato

Bağlantı Yapıları – 2/4

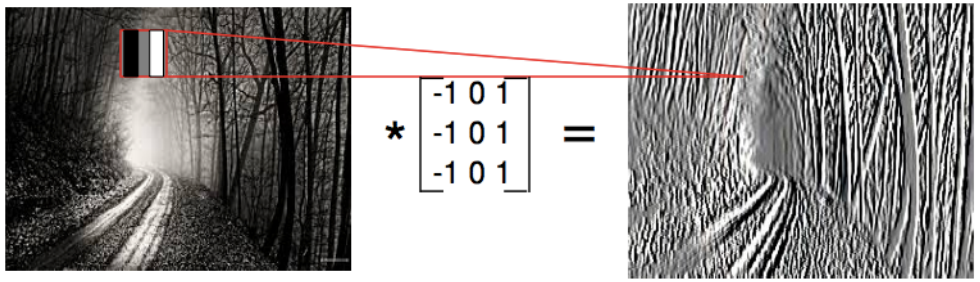
Yerel Bağlı

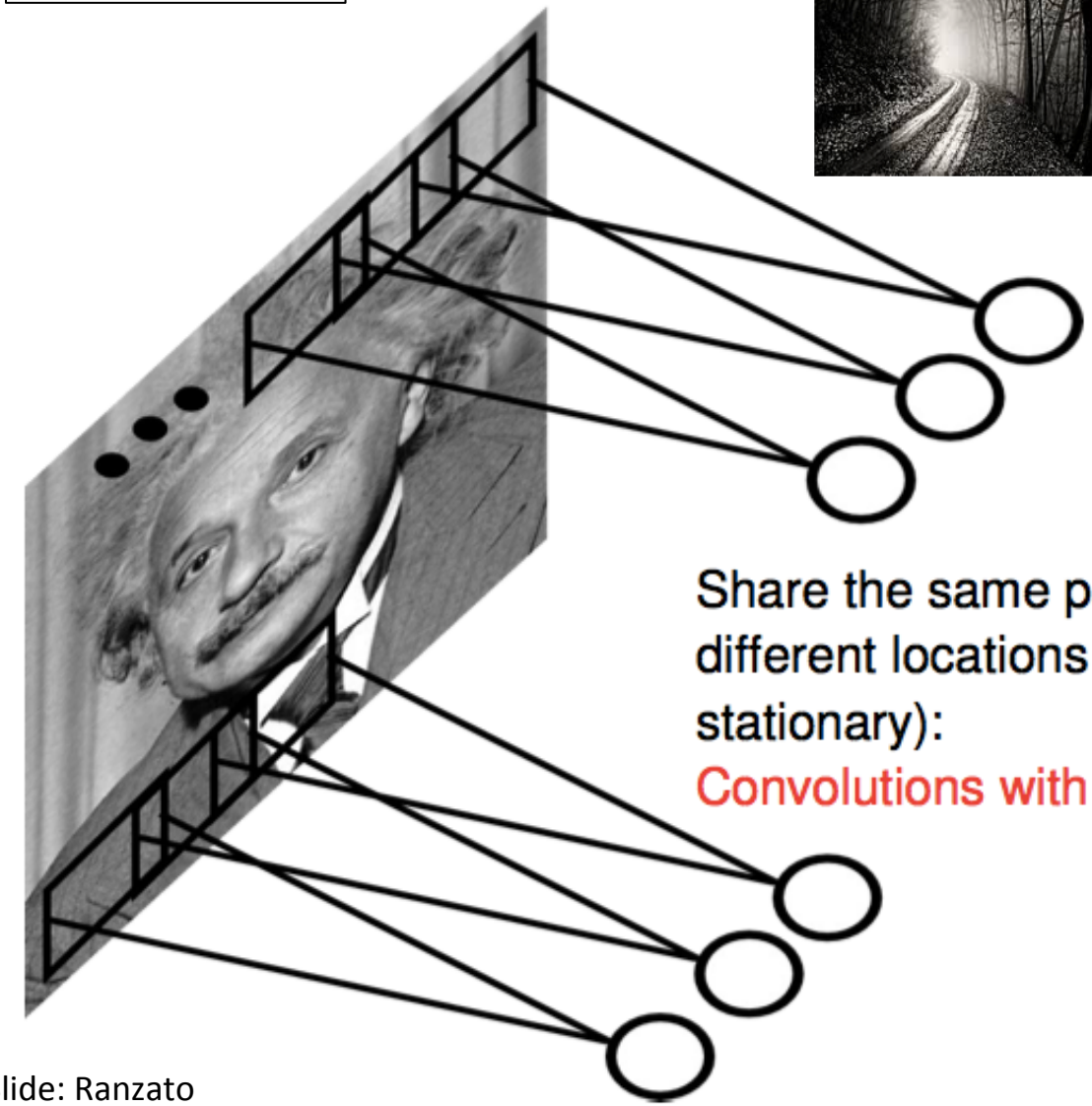


Slide: Ranzato

Bağlantı Yapıları – 3/4

Evrişimsel



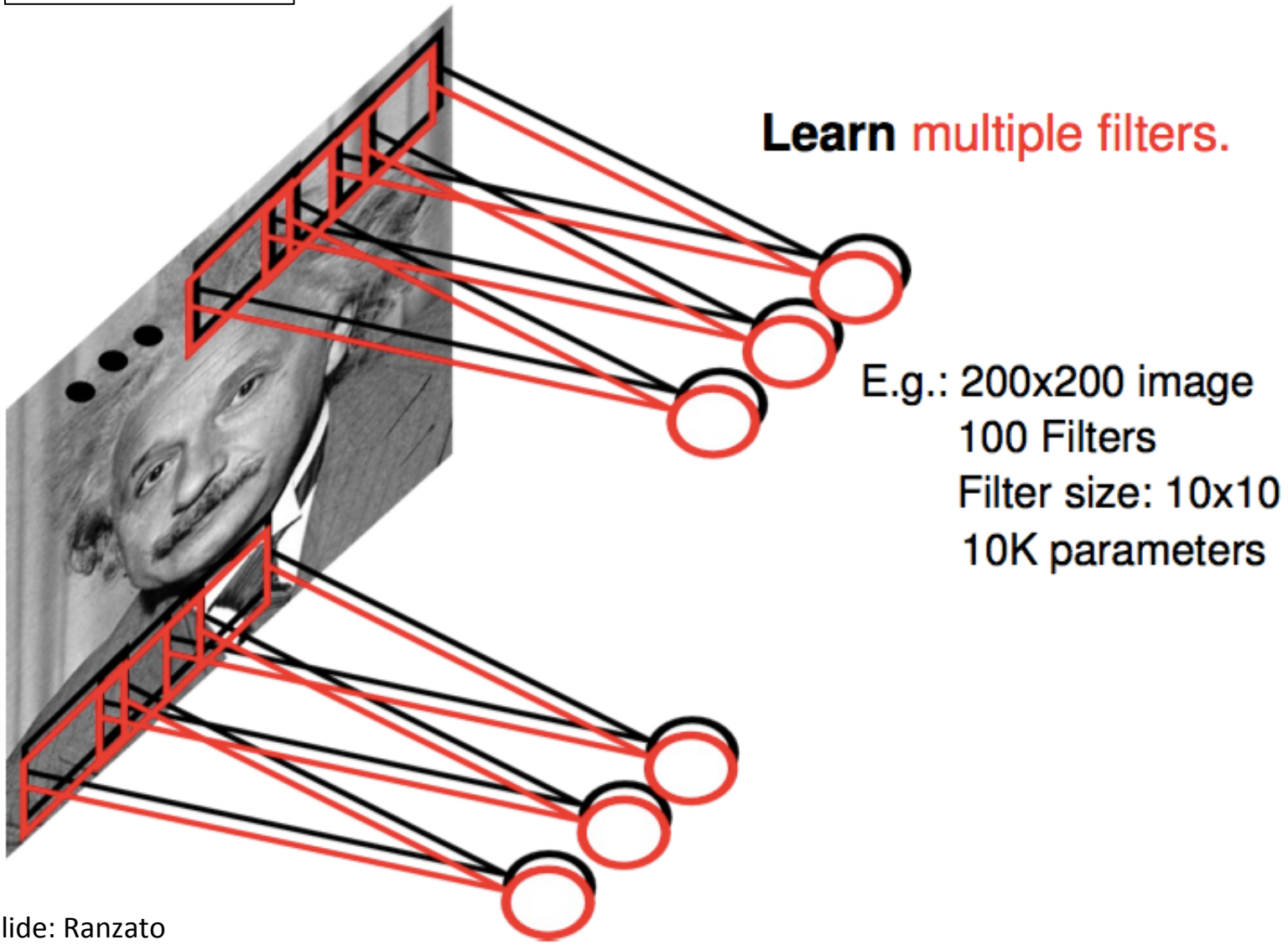


Share the same parameters across different locations (assuming input is stationary):
Convolutions with learned kernels

Slide: Ranzato

Bağlantı Yapıları – 4/4

Evrişimsel

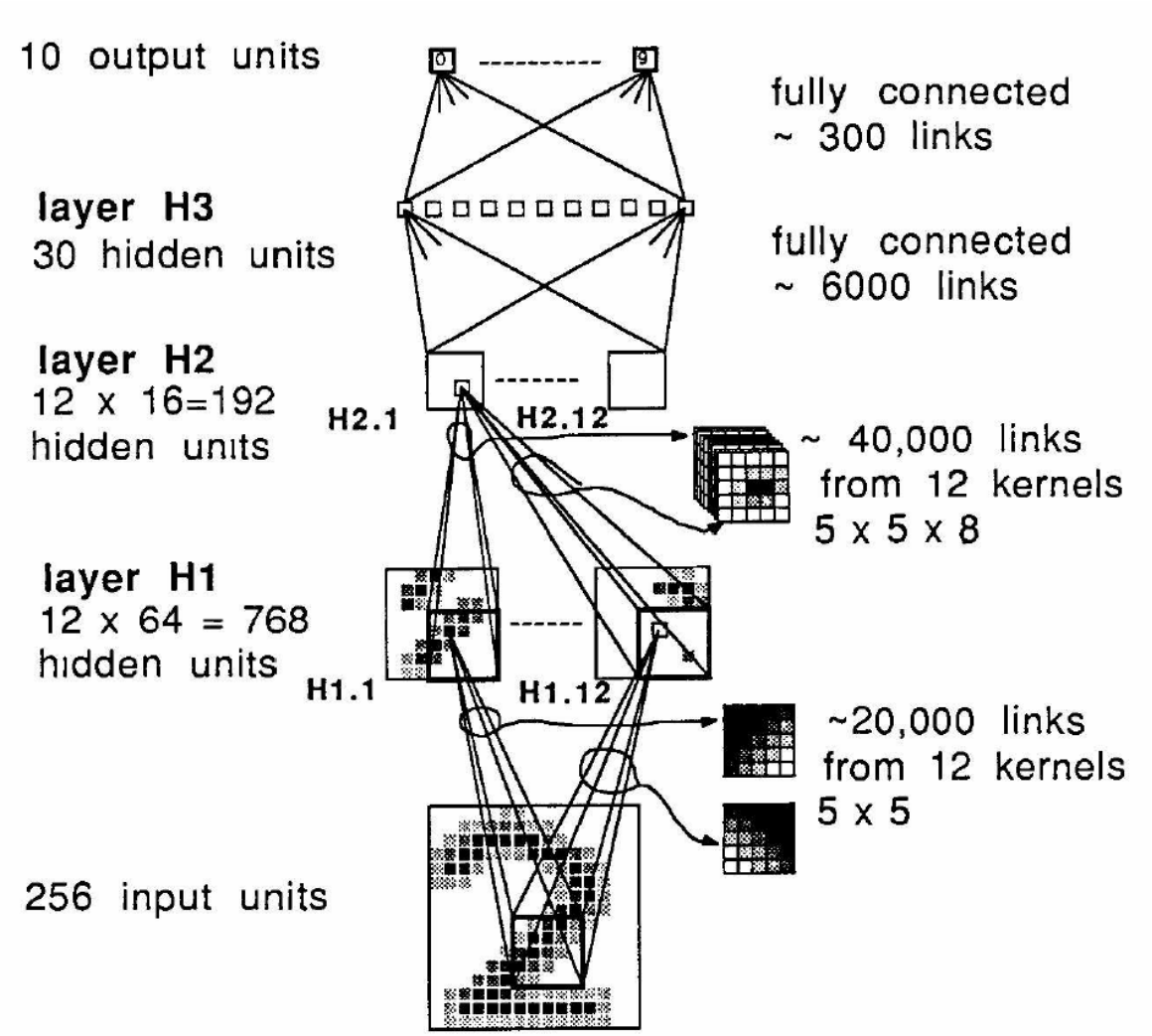


Slide: Ranzato

ESA: İlk Zamanlar – 1/2

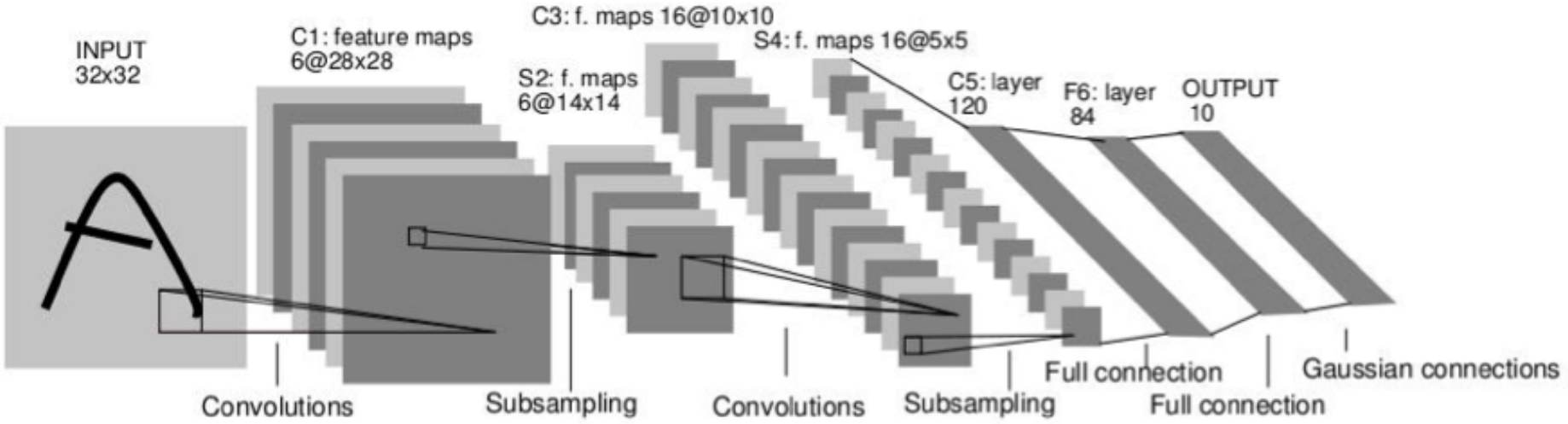
LeCun et al. 1989

Elyazısı rakam karakterleri
7291 eğitim örneği
2007 test örneği
Üç katman
64460 bağlantı
9760 serbest parametre
Başarım
5.0% test hatası
Karşılaştırma
Tam-bağlı tek katman
40 units,
8.1% test error



ESA: İlk Zamanlar – 2/2

LeCun et al. 1998



- Özel bağlantı yapılı sinir ağı -> **Evrişimsel**
- İleriye Beslemeli (feed-forward), her adımda:
  - Girdiyi evriştir
  - "ReLU" uygula: $f(u) = \max(0,u)$
  - "Pooling" (local max)
- Her adımın çıktısı alt örnekle ve bir sonraki adıma besle
- Uzmanlı eğitim, "backpropagation"

LeCun-Tarzı ESA'ların Başarıları

Handwritten text/digits
- MNIST (0.17% error [Ciresan et al. 2011])
- Arabic & Chinese [Ciresan et al. 2012]

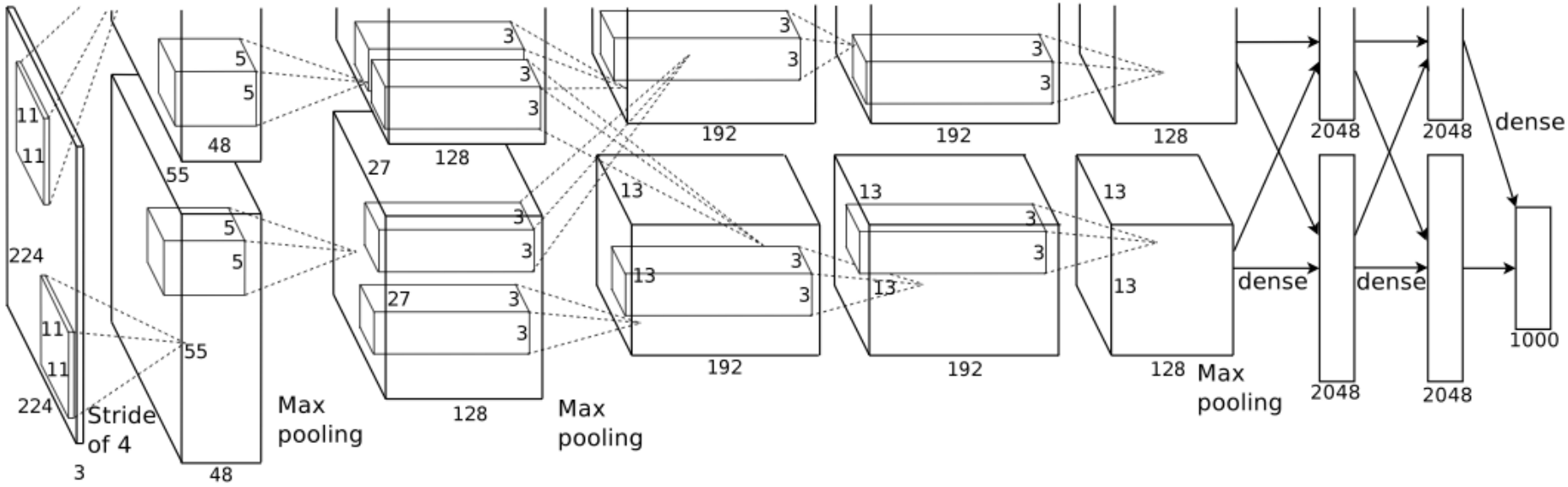
Simpler recognition benchmarks
- CIFAR-10 (9.3% error [Wan et al. 2013])
- Traffic sign recognition
  - 0.56% error vs 1.16% for humans [Ciresan et al. 2011]

But (until recently) less good at more complex datasets
- E.g. Caltech-101/256 (few training examples)

Slide: Fergus

Yeni Dönem ESA'ları

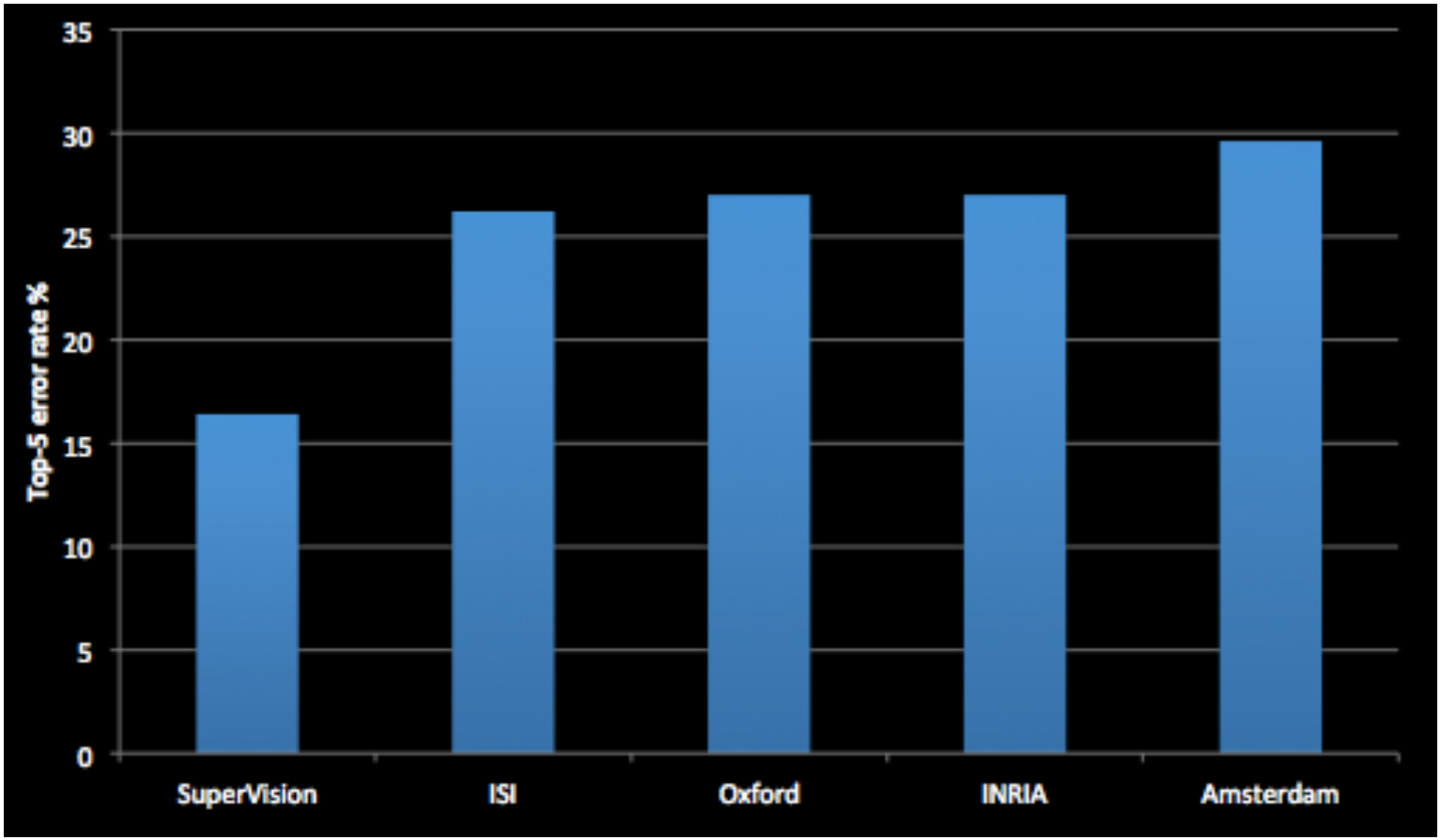
Krizhevsky et al. 2012 – AlexNet



- LeCun'98 ile aynı model ama çok daha büyük:
  - 8 katman, 7'si saklı: 650K nörons, 60M parametre
  - Daha fazla veri (1M vs 1K images)
  - GPU (CPU'ya göre 50x hız kazanımı): Bir hafta boyunca iki GPU kullanılarak eğitildi
  - Daha iyi düzenlileştirme (Drop-Out)

AlexNet ve ImageNet 2012

- 1000 sınıf, 1.3M etiketli eğitim örneği
- 16.4% top-5 hatası (en yakın rakibi 26.2%)



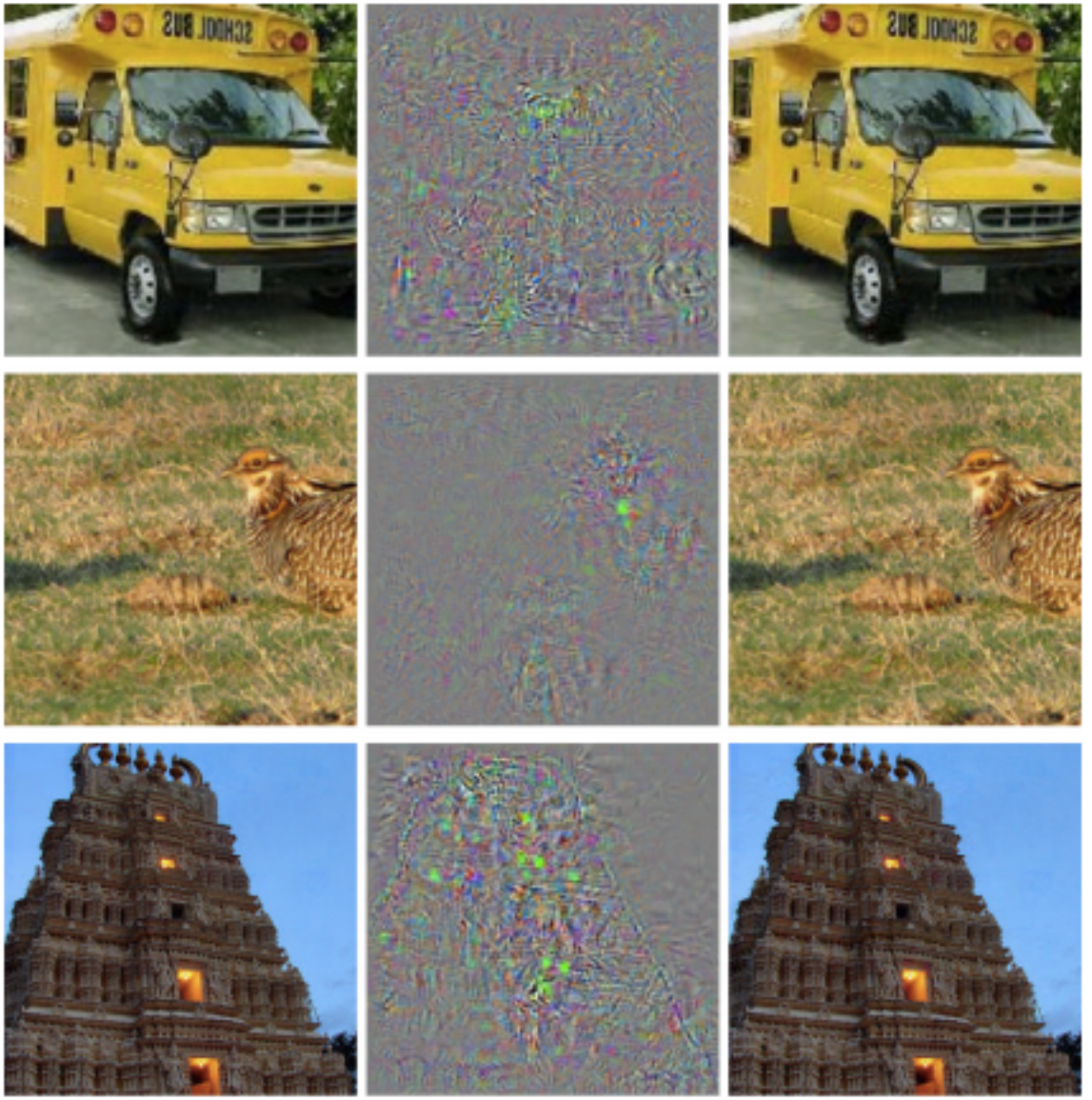
Ama şu da var:

Deep Neural Networks are Easily Fooled:
High Confidence Predictions for Unrecognizable Images

Anh Nguyen
University of Wyoming
anguyen8@uwyo.edu

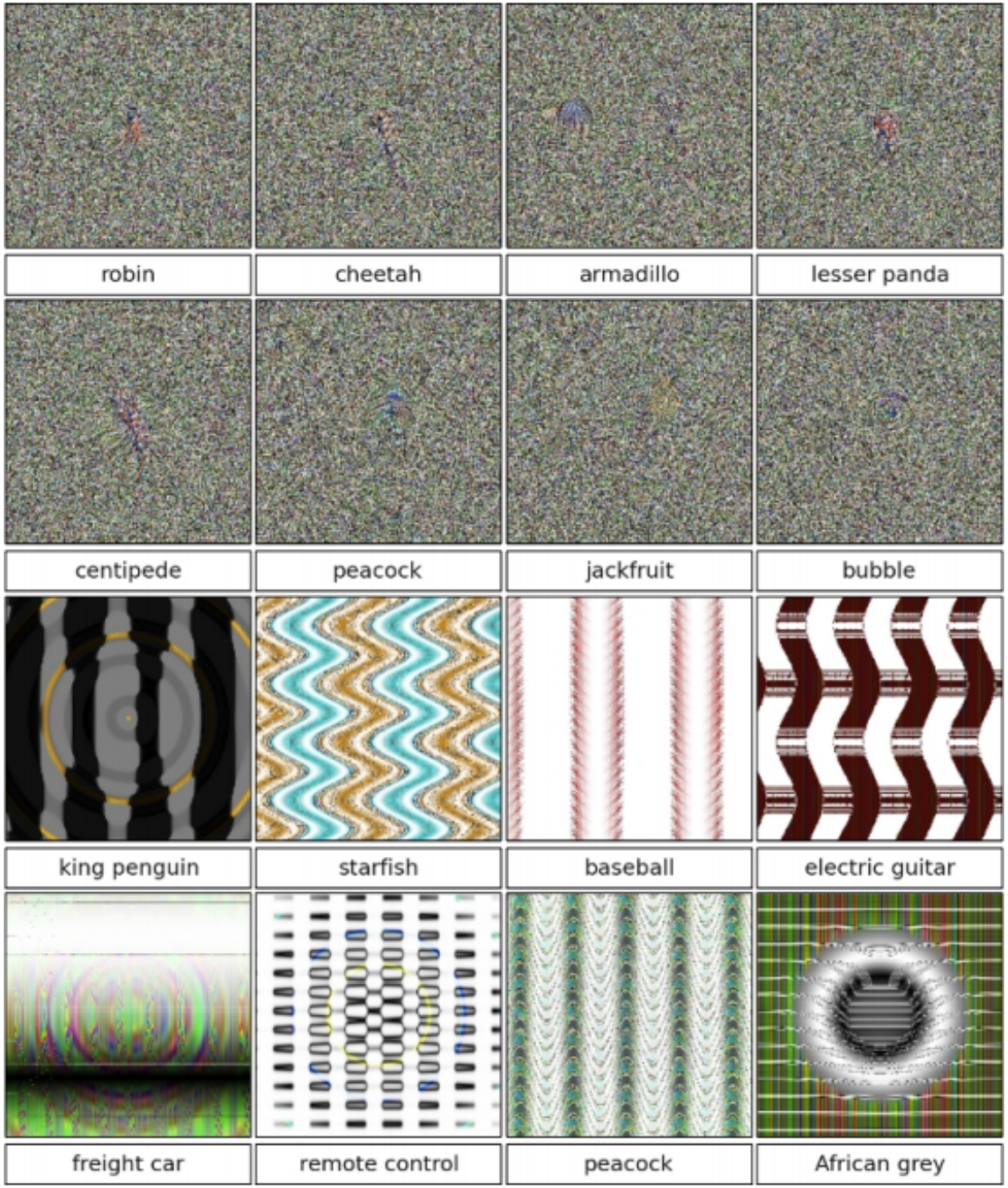
Jason Yosinski
Cornell University
yosinski@cs.cornell.edu

Jeff Clune
University of Wyoming
jeffclune@uwyo.edu



High C **mages**

Anh Nguy ff Clune
University of W y of Wyoming
anguyen8@uwy ne@uwyo.edu



High C **mages**

Anh Nguy ff Clune
University of W y of Wyoming
anguyen8@uwy ne@uwyo.edu

Modern ESA Mimarisi

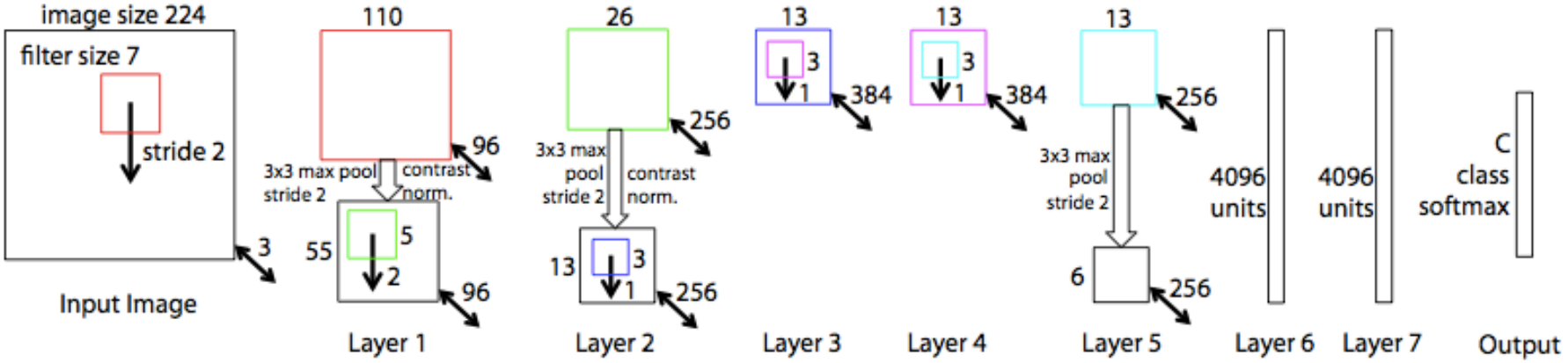
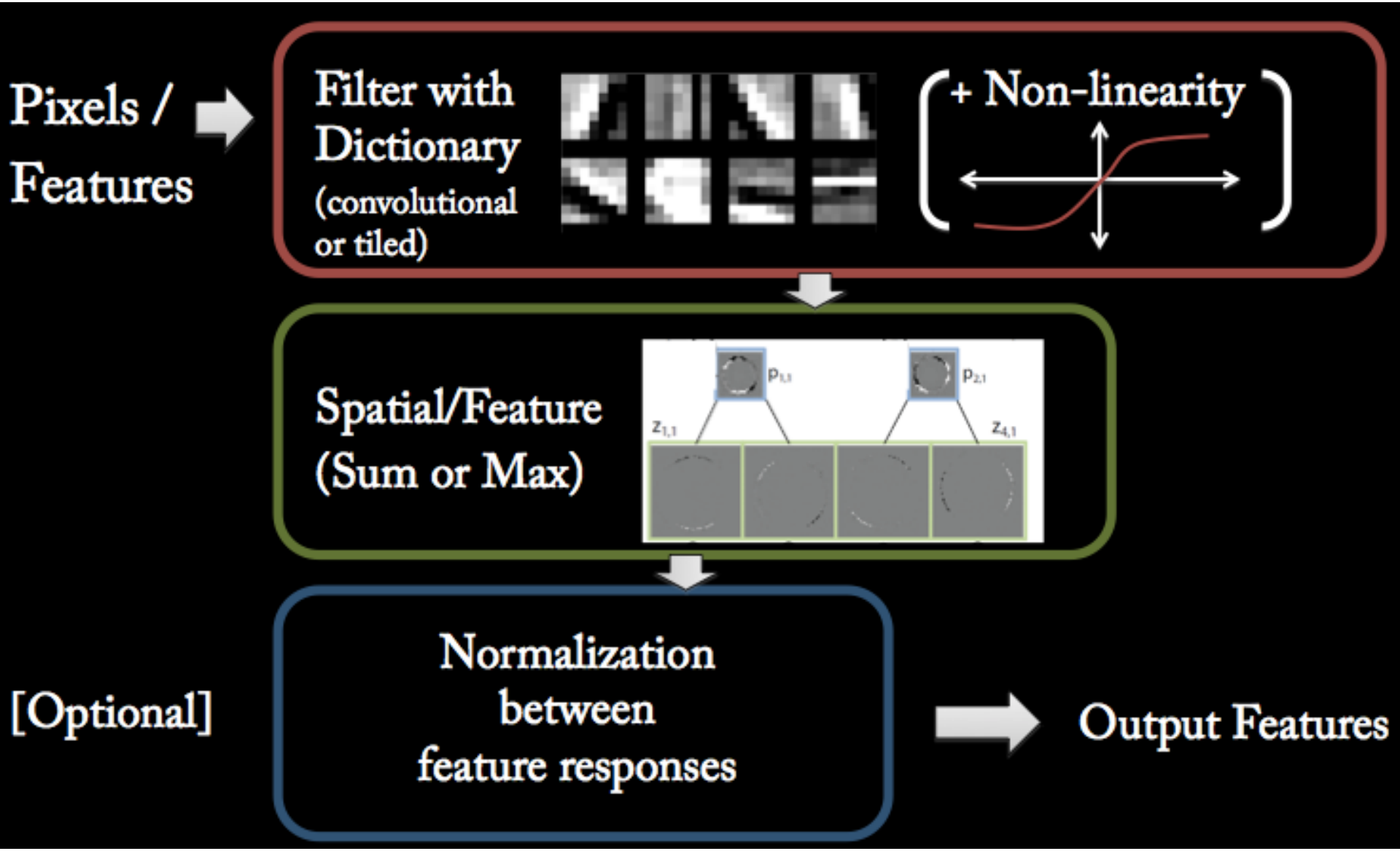


Fig. 3. Architecture of our 8 layer convnet model. A 224 by 224 crop of an image (with 3 color planes) is presented as the input. This is convolved with 96 different 1st layer filters (red), each of size 7 by 7, using a stride of 2 in both x and y. The resulting feature maps are then: (i) passed through a rectified linear function (not shown), (ii) pooled (max within 3x3 regions, using stride 2) and (iii) contrast normalized across feature maps to give 96 different 55 by 55 element feature maps. Similar operations are repeated in layers 2,3,4,5. The last two layers are fully connected, taking features from the top convolutional layer as input in vector form ($6 \cdot 6 \cdot 256 = 9216$ dimensions). The final layer is a C-way softmax function, C being the number of classes. All filters and feature maps are square in shape.

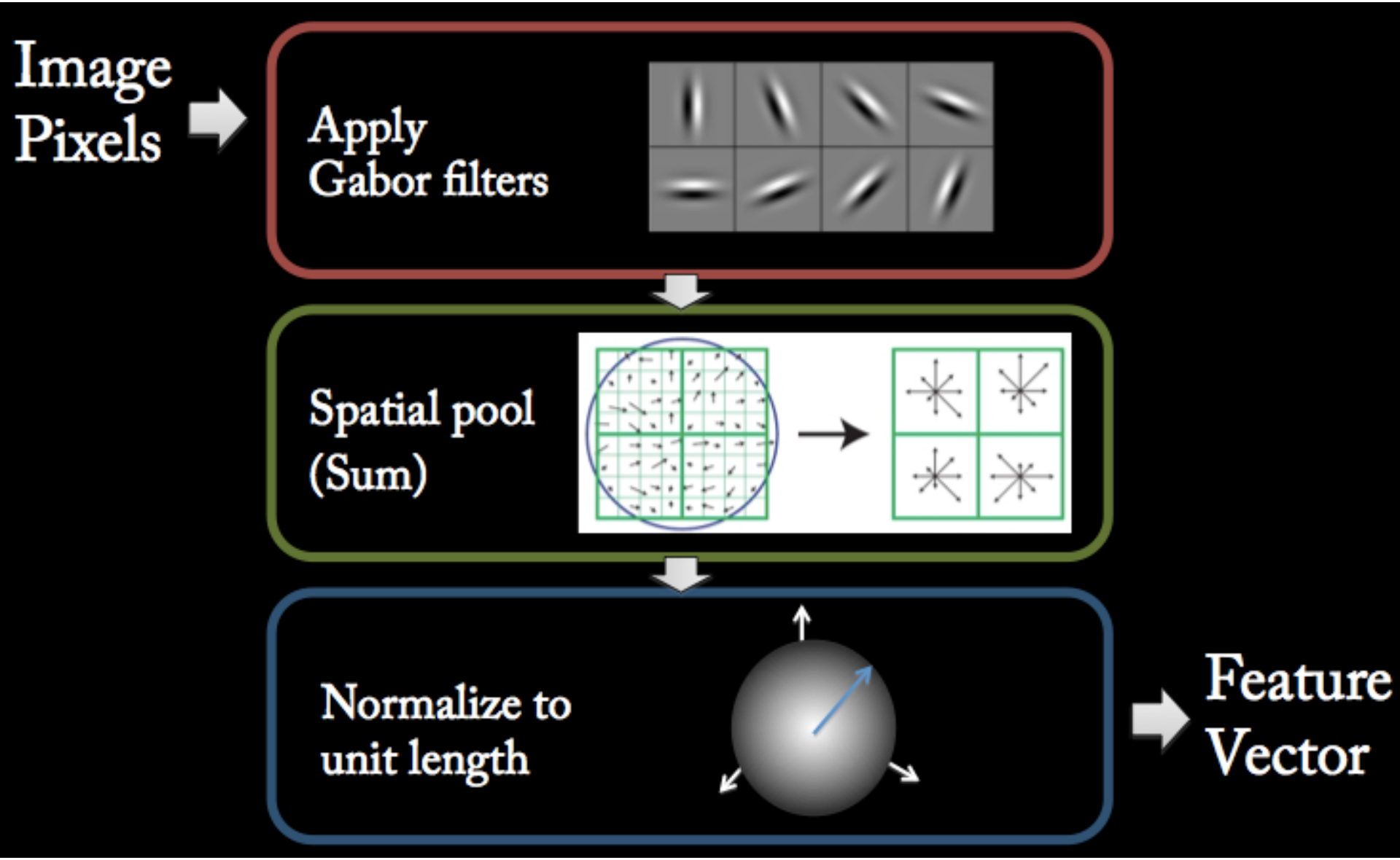
Zeiler and Fergus 2013

Bir ESA Katmanının Bileşenleri



Slide: Fergus

SIFT ile Karşılaştırma



Slide: Fergus

Süzgeçleme

- Convolutional
  - Dependencies are local
  - Translation equivariance
  - Tied filter weights (few params)
  - Stride 1,2,… (faster, less mem.)

Input

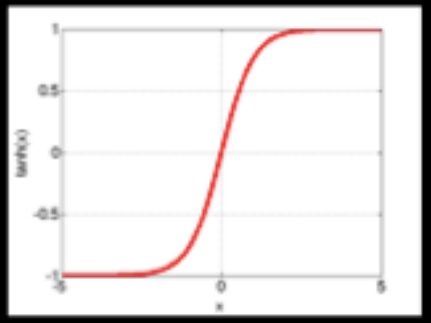
Feature Map

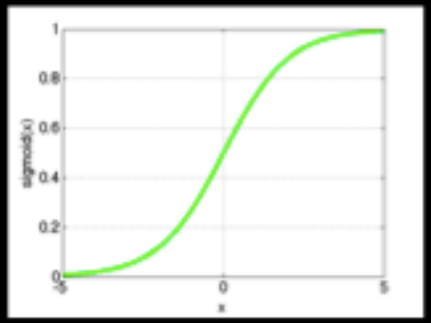
Slide: Fergus

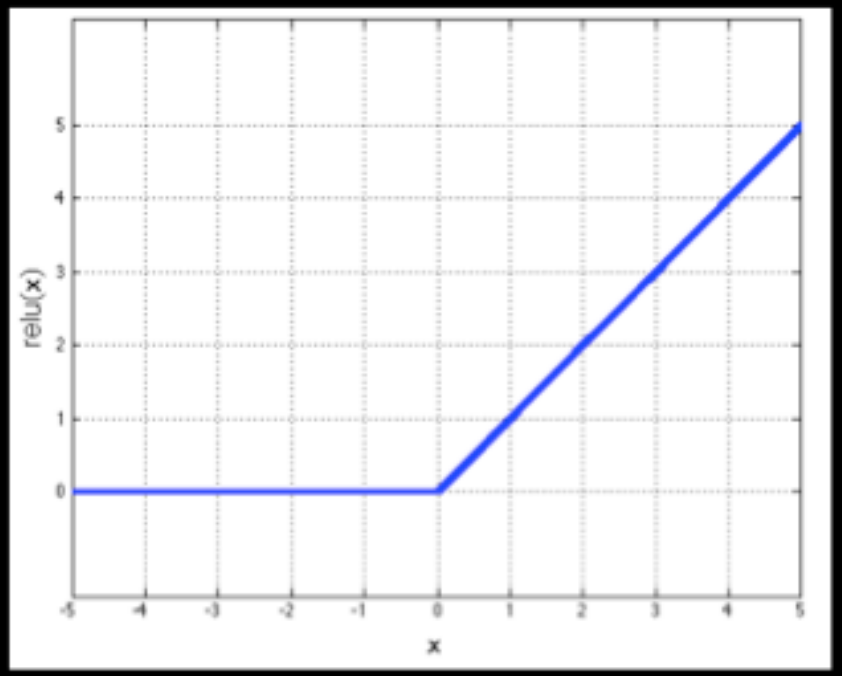
“Non-Linearity”

- Non-linearity
  - Per-feature independent
  - Tanh
  - Sigmoid: 1/(1+exp(-x))
  - Rectified linear
    - Simplifies backprop
    - Makes learning faster
    - Avoids saturation issues

    → Preferred option



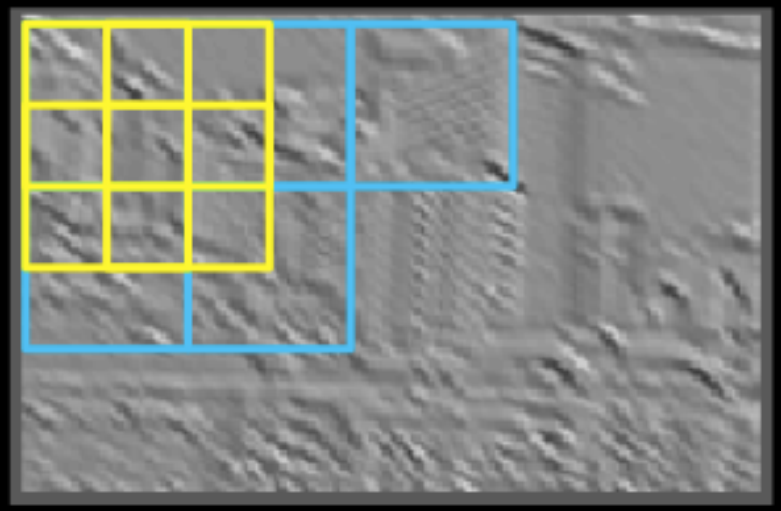




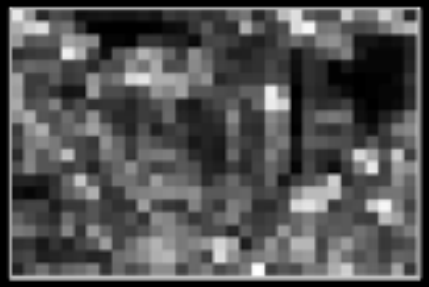
Slide: Fergus

“Pooling” – 1/2

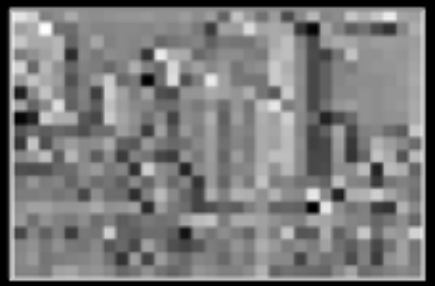
- Spatial Pooling
  - Non-overlapping / overlapping regions
  - Sum or max



Max



Sum



Slide: Fergus

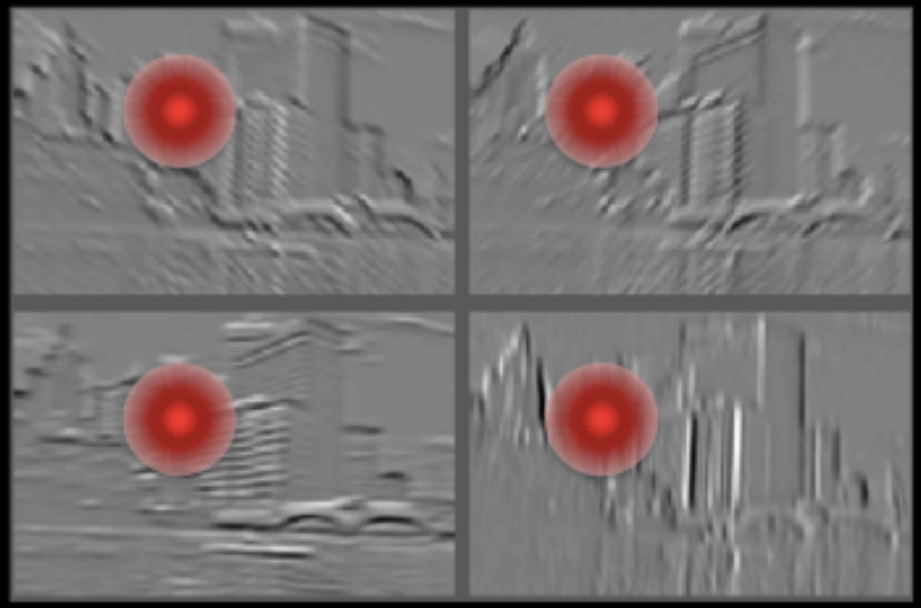
“Pooling” – 2/2

- Pooling across feature groups
  - Additional form of inter-feature competition

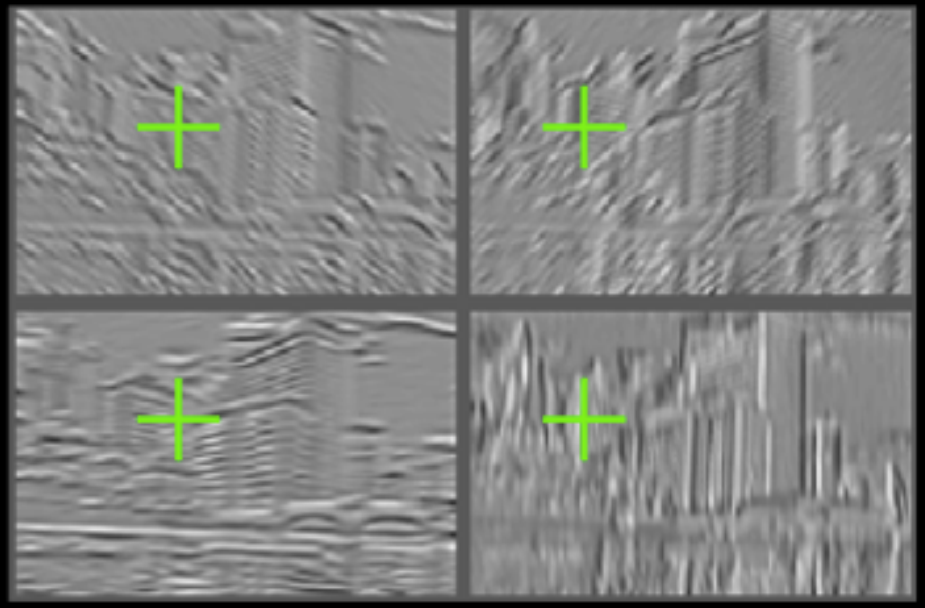
Slide: Fergus

Normalizasyon

- Contrast normalization (between/across feature maps)
  - Local mean = 0, local std. = 1, "Local" → 7x7 Gaussian
  - Equalizes the features maps



Feature Maps

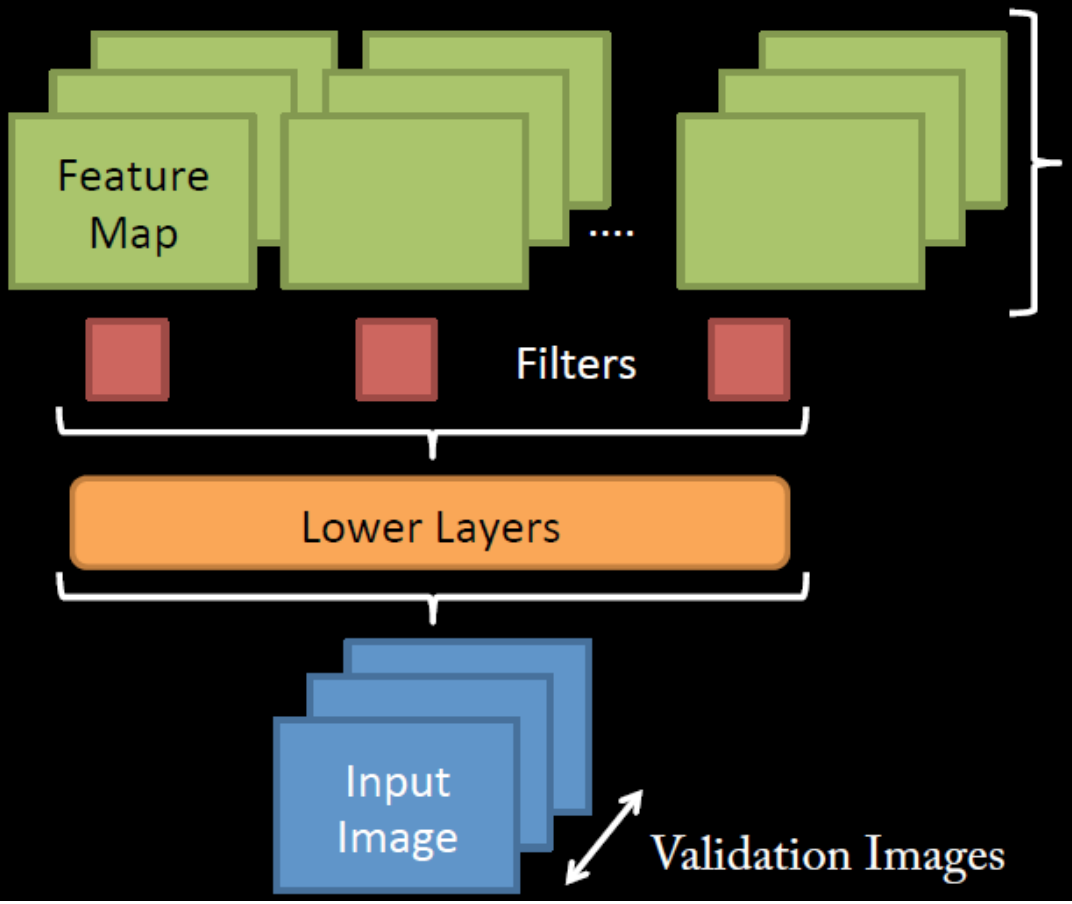


Feature Maps
After Contrast Normalization

Slide: Fergus

ESA'ları Görselleştirmek – 1/3

- Use ImageNet 2012 validation set [Zeiler and Fergus. arXiv'13]
- Push each image through network

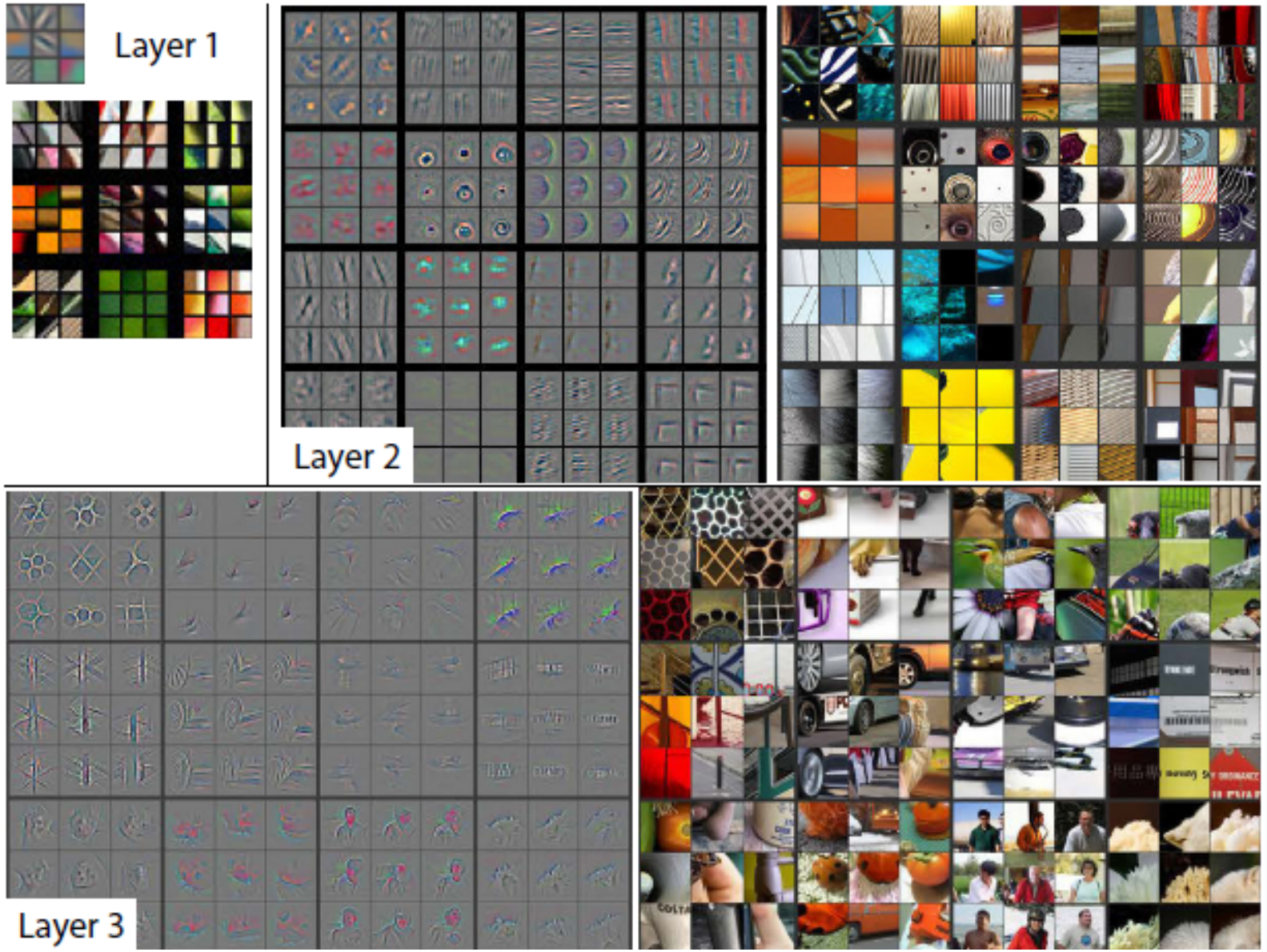


- Take max activation from feature map associated with each filter
- Use Deconvnet to project back to pixel space
- Use pooling "switches" peculiar to that activation

Slide: Fergus

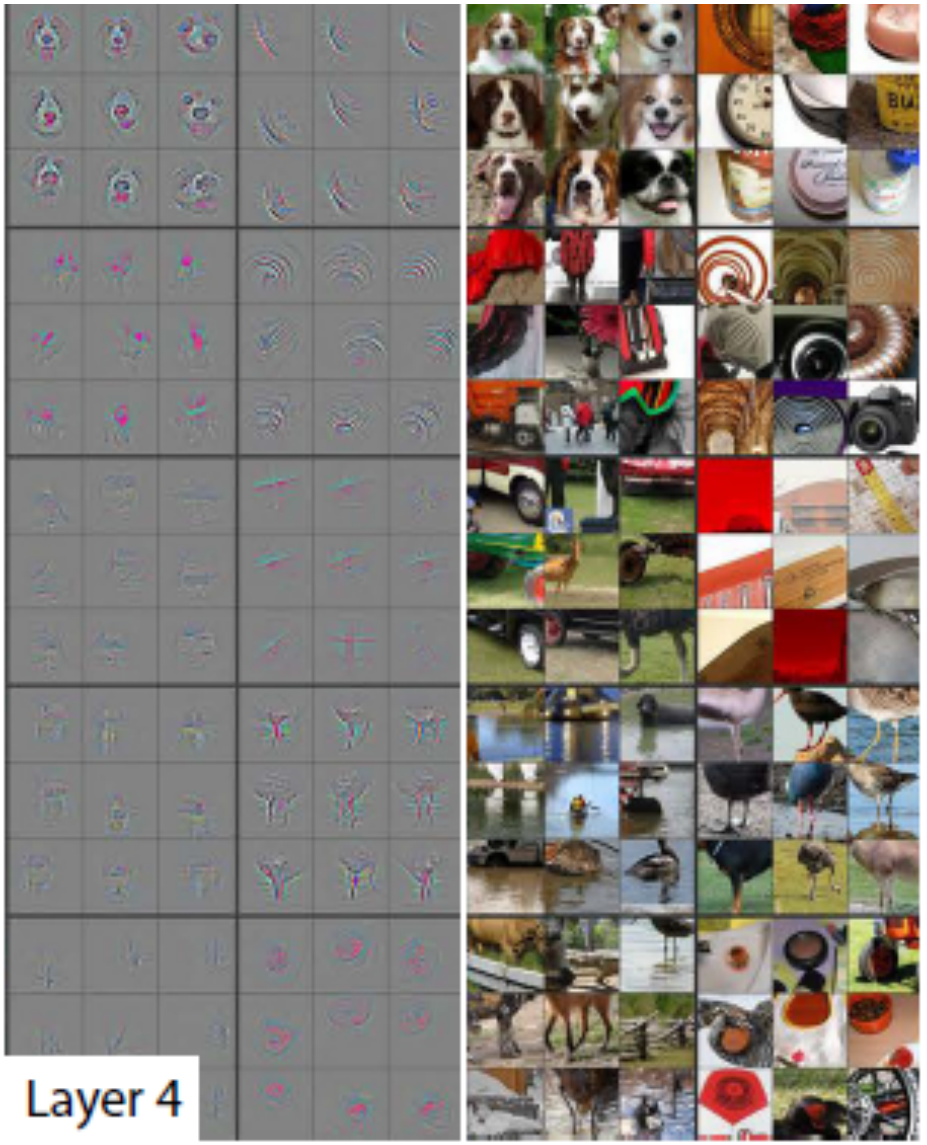
ESA'ları Görselleştirmek – 2/3

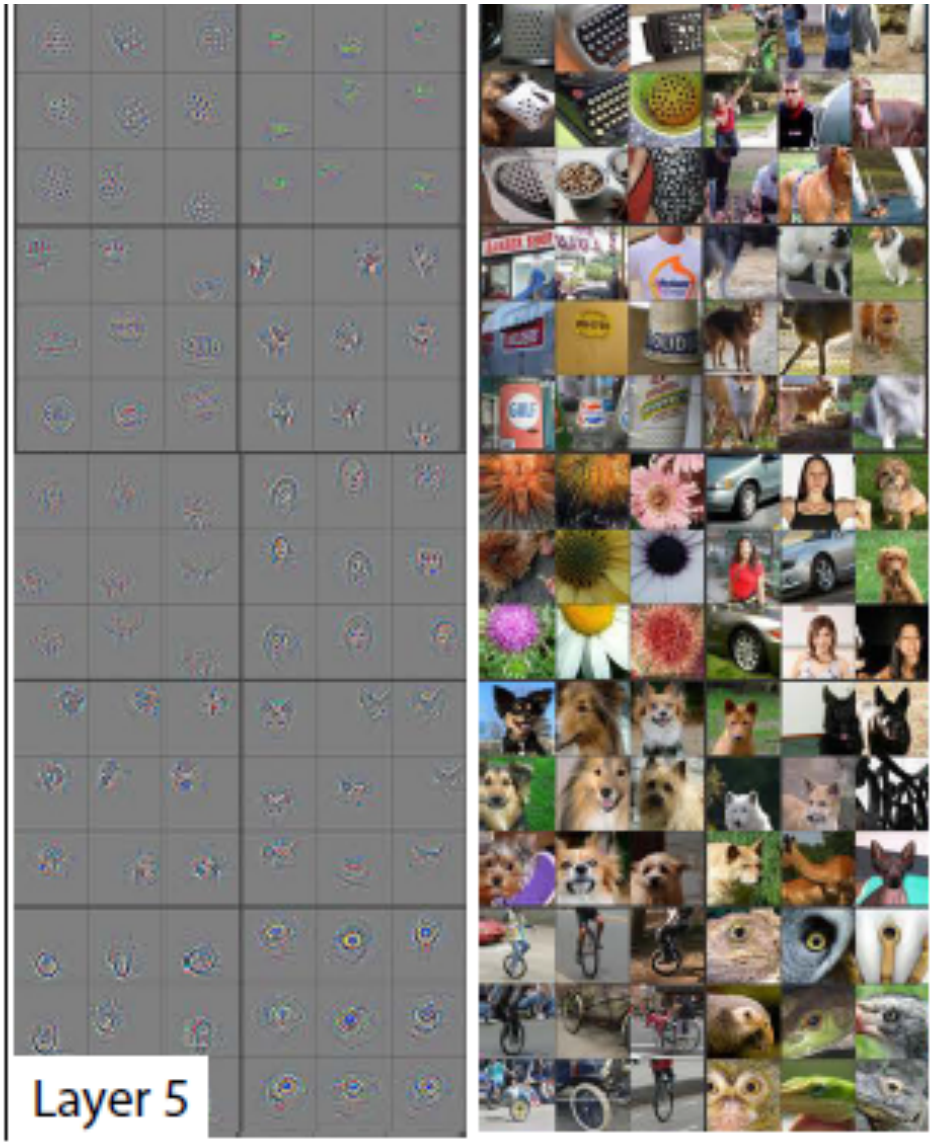
Zeiler and Fergus 2013



ESA'ları Görselleştirmek – 3/3

Zeiler and Fergus 2013





“Yeni” ESA Eğitim Stratejisi – 1/2

Yüksek kapasiteli (*) bir ESA’nın başarımı çok sayıda etiketli eğitim örneği olmasına bağlıdır.

AlexNet: 60 milyon parametre

ImageNet: 1.3 milyon örnek!

(*) Large-capacity = Millions of parameters

“Yeni” ESA Eğitim Stratejisi – 1/2

Yüksek kapasiteli (*) bir ESA’nın başarımı çok sayıda etiketli eğitim örneği olmasına bağlıdır.

AlexNet: 60 milyon parametre

ImageNet: 1.3 milyon örnek!

Peki elde o kadar örnek yoksa?
Mesela Caltech-101?

(*) Yüksek kapasite = Milyonlarca parametre

“Yeni” ESA Eğitim Stratejisi – 2/2

DECAF: Donahue et al. 2013

(1) AlexNet-tarzı bir ESA’yı ImageNet’te eğit, amaç imge sınıflandırma olsun

(2) Çıktı katmanını at, diğerlerini tut

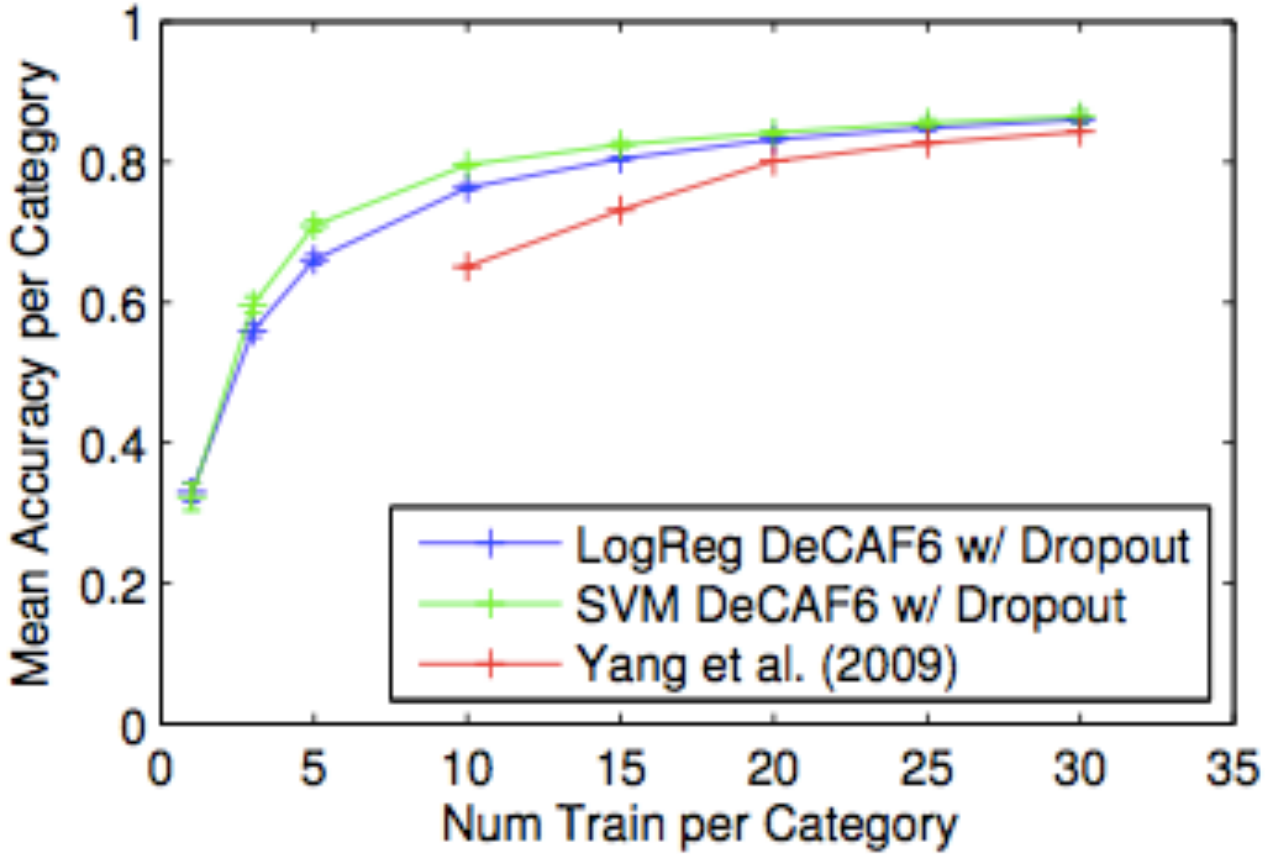
(3) ESA’nin kalan kısmını öznitelik çıkarıcı olarak kullan

(4) Çıkarılan öznitelikleri başka bir veri kümesinde ve başka bir görsel tanıma problemi için kullan

Buna kısmen “Transfer Learning” deniyor!

DECAF Performansı – 1/4

Caltech-101: Nesne Tanıma



Donahue et al. 2013

DECAF Performansı – 2/4

"Domain Adaption"

	Amazon → Webcam			Dslr → Webcam		
	SURF	$DeCAF_6$	$DeCAF_7$	SURF	$DeCAF_6$	$DeCAF_7$
Logistic Reg. (S)	9.63 ± 1.4	48.58 ± 1.3	53.56 ± 1.5	24.22 ± 1.8	88.77 ± 1.2	87.38 ± 2.2
SVM (S)	11.05 ± 2.3	52.22 ± 1.7	53.90 ± 2.2	38.80 ± 0.7	91.48 ± 1.5	89.15 ± 1.7
Logistic Reg. (T)	24.33 ± 2.1	72.56 ± 2.1	74.19 ± 2.8	24.33 ± 2.1	72.56 ± 2.1	74.19 ± 2.8
SVM (T)	51.05 ± 2.0	78.26 ± 2.6	78.72 ± 2.3	51.05 ± 2.0	78.26 ± 2.6	78.72 ± 2.3
Logistic Reg. (ST)	19.89 ± 1.7	75.30 ± 2.0	76.32 ± 2.0	36.55 ± 2.2	92.88 ± 0.6	91.91 ± 2.0
SVM (ST)	23.19 ± 3.5	80.66 ± 2.3	79.12 ± 2.1	46.32 ± 1.1	$\mathbf{94.79 \pm 1.2}$	92.96 ± 2.0
Daume III (2007)	40.26 ± 1.1	$\mathbf{82.14 \pm 1.9}$	81.65 ± 2.4	55.07 ± 3.0	91.25 ± 1.1	89.52 ± 2.2
Hoffman et al. (2013)	37.66 ± 2.2	80.06 ± 2.7	80.37 ± 2.0	53.65 ± 3.3	93.25 ± 1.5	91.45 ± 1.5
Gong et al. (2012)	39.80 ± 2.3	75.21 ± 1.2	77.55 ± 1.9	39.12 ± 1.3	88.40 ± 1.0	88.66 ± 1.9
Chopra et al. (2013)		58.85			78.21	

Donahue et al. 2013

DECAF Performansı – 3/4

UCSD-Caltech Bird: Alt Kategori Tanıma

Method	Accuracy
$DeCAF_6$	58.75
DPD + $DeCAF_6$	**64.96**
DPD (Zhang et al., 2013)	50.98
POOF (Berg & Belhumeur, 2013)	56.78

Donahue et al. 2013

DECAF Performansı – 4/4

SUN-397: Sahne Tanıma

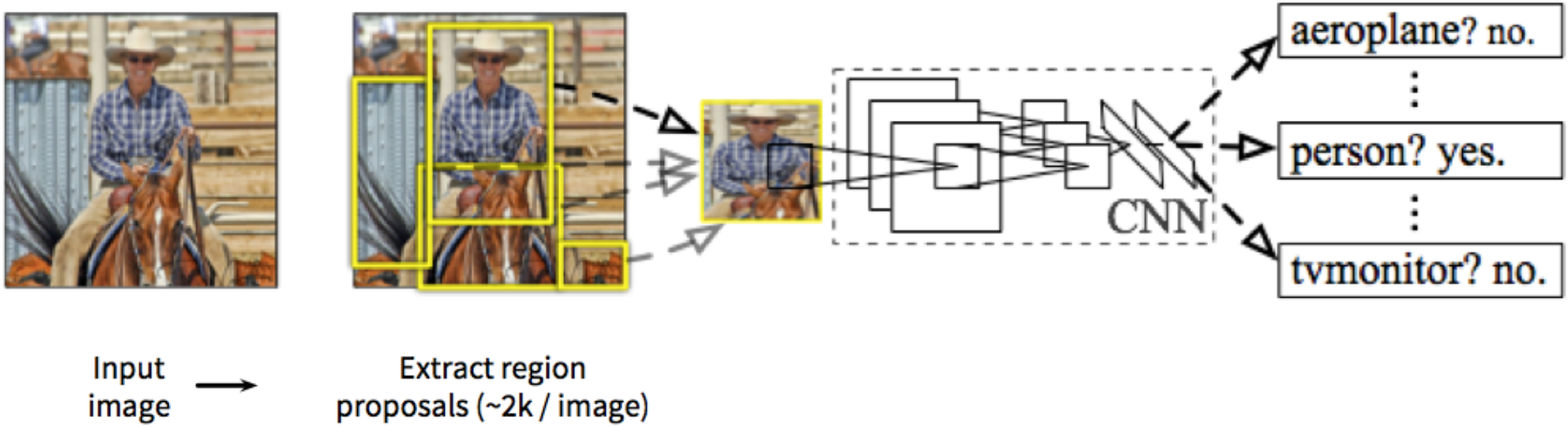
	$DeCAF_6$	$DeCAF_7$
LogReg	$\mathbf{40.94 \pm 0.3}$	40.84 ± 0.3
SVM	39.36 ± 0.3	40.66 ± 0.3
Xiao et al. (2010)	38.0	

Table 3. Average accuracy per class on SUN-397 with 50 training samples and 50 test samples per class, across two hidden layers of the network and two classifiers. Our result from the training protocol/classifier combination with the best validation accuracy – Logistic Regression with $DeCAF_7$ – is shown in **bold**.

Donahue et al. 2013

ESA'larla Nesne Bulma – 1/3

R-CNN Pipeline [Girschik et al. 2014]



- **Proposal-method agnostic, many choices**
  - Selective Search [Uijlings et al. 2013] (Used in this work)
  - Objectness [Alexe et al. 2012]
  - Category independent object proposals [Endres & Hoiem 2010]

ESA'larla Nesne Bulma – 2/3

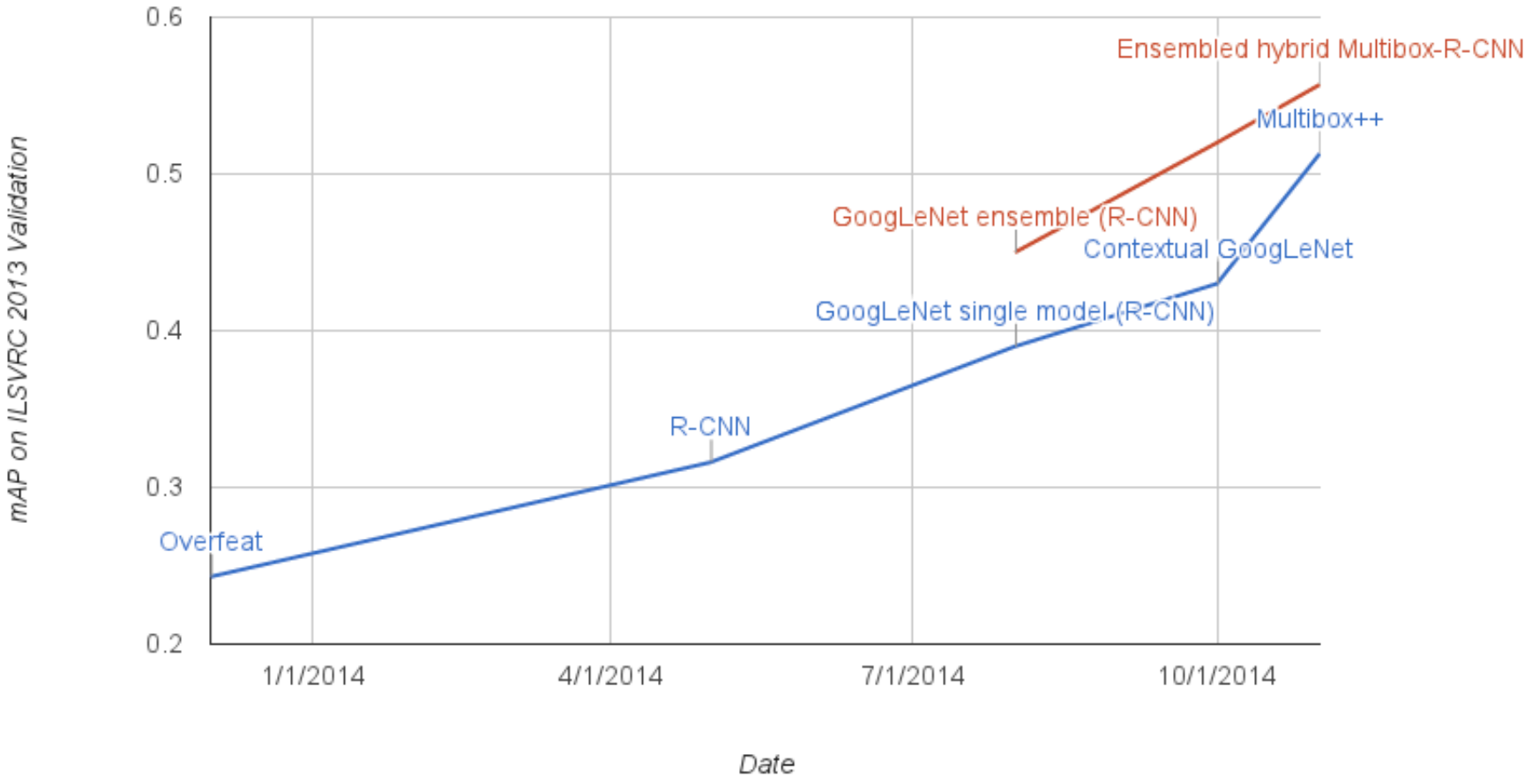
R-CNN Performance

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [18]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [34]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [36]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [16]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

Table 1: Detection average precision (%) on VOC 2010 test. R-CNN is most directly comparable to UVA and Regionlets since all methods use selective search region proposals. Bounding box regression (BB) is described in Section 3.4. At publication time, SegDPM was the top-performer on the PASCAL VOC leaderboard. †DPM and SegDPM use context rescoring not used by the other methods.

ESA'larla Nesne Bulma – 3/3

ILSVRC Nesne Bulma Yarışması Sonuçları



Popüler Yazılım Kütüphaneleri – 1/2

- Theano + PyLearn2
- Torch & OverFeat
- Caffe
- Cuda-ConvNet
- DeepLearning4J

Niteliksel Karşılaştırma [Caffe paper]

Framework	License	Core language	Binding(s)	CPU	GPU	Open source	Training	Pretrained models	Development
Caffe	BSD	C++	Python, MATLAB	✓	✓	✓	✓	✓	distributed
cuda-convnet [7]	unspecified	C++	Python		✓	✓	✓		discontinued
Decaf [2]	BSD	Python		✓		✓	✓	✓	discontinued
OverFeat [9]	unspecified	Lua	C++,Python	✓				✓	centralized
Theano/Pylearn2 [4]	BSD	Python		✓	✓	✓	✓		distributed
Torch7 [1]	BSD	Lua		✓	✓	✓	✓		distributed

Popüler Yazılım Kütüphaneleri – 2/2

Performans

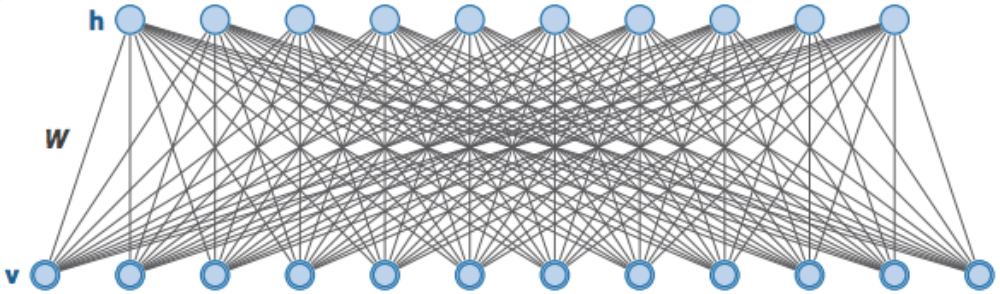
forward + backprop (wrt input and weights)

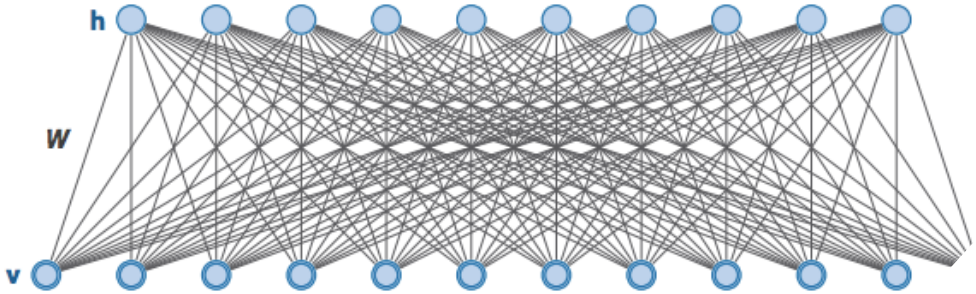
Original Library	Class/Function Benchmarked	Time (ms)	forward (ms)	backward (ms)	
fbfft	SpatialConvolutionCuFFT	**256**	**101**	**155**	Facebook (LeCun)
cuda-convnet2 *	ConvLayer	977	201	776	Berkeley (Krizhevsky)
cuda-convnet**	pylearn2.cuda_convnet	1077	312	765	Berkeley (Krizhevsky)
CuDNN R2 *	cudnn.SpatialConvolution	1019	269	750	NVIDIA
Theano	CorrMM	1225	407	818	Bengio
Caffe	ConvolutionLayer	1231	396	835	Berkeley
Torch-7	SpatialConvolutionMM	1265	418	877	Consortium (Google, Facebook, ...)
DeepCL	ConvolutionLayer	6280	2648	3632	
*cherry-picking****	*best per layer*	*235*	*79*	*155*	

* https://github.com/soumith/convnet-benchmarks [Soumith Chintala]

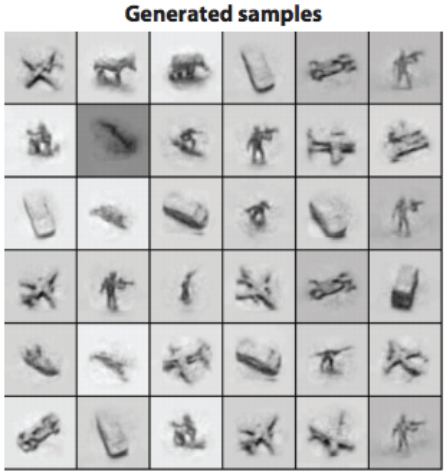
Sırada Ne Var?

Ağırlıkları ve yapıyı öğren





Yeni örnekler üret (sentezle)



Bitirirken…

J. W. Goethe (1749-1832)

“Deha uygulamadadır.”

“İyi bir teoriden daha pratik hiçbir şey yoktur.”

Kurt Lewin (1890-1947)