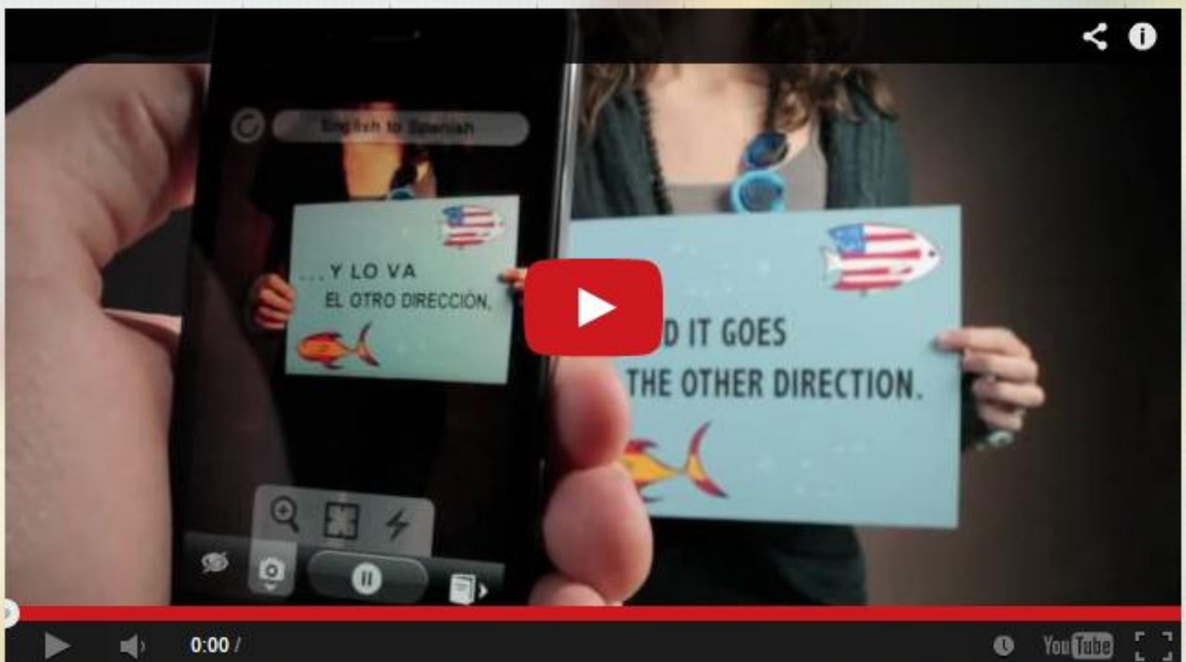
Pattern Recognition:
Feature Engineering and (Deep) Feature Learning

DungDuc NGUYEN, Ph.D.
Institute of Information Technology, VAST

Quest Visual

English

WORD LENS



See the world in your language.

Word Lens translates printed words from one language to another with your smartphone's video camera, in real time. No network connection needed!

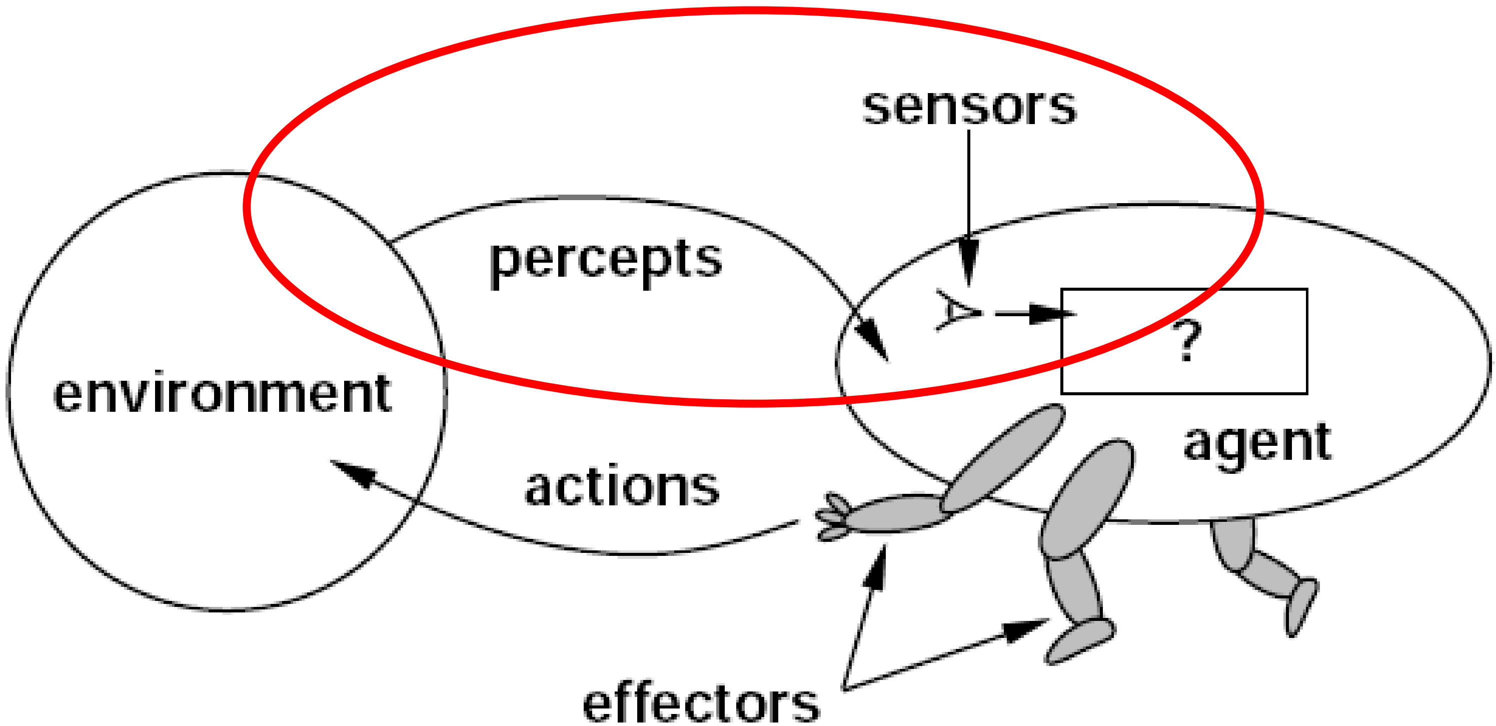
GET IT ON GLASS

GET IT ON Google play

Available on the App Store

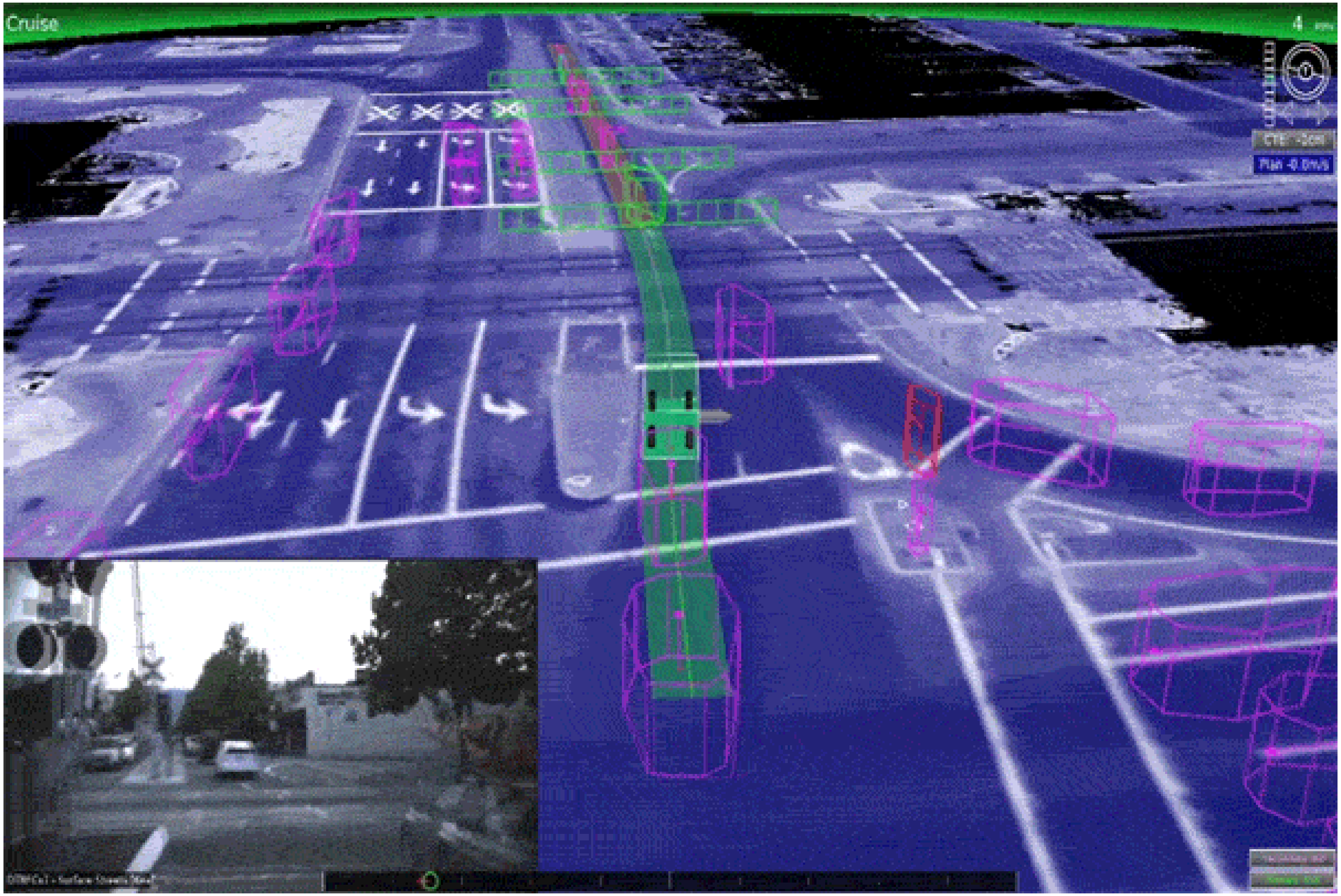
Bloomberg   WIRED   The New York Times   TechCrunch   The Economist

AI

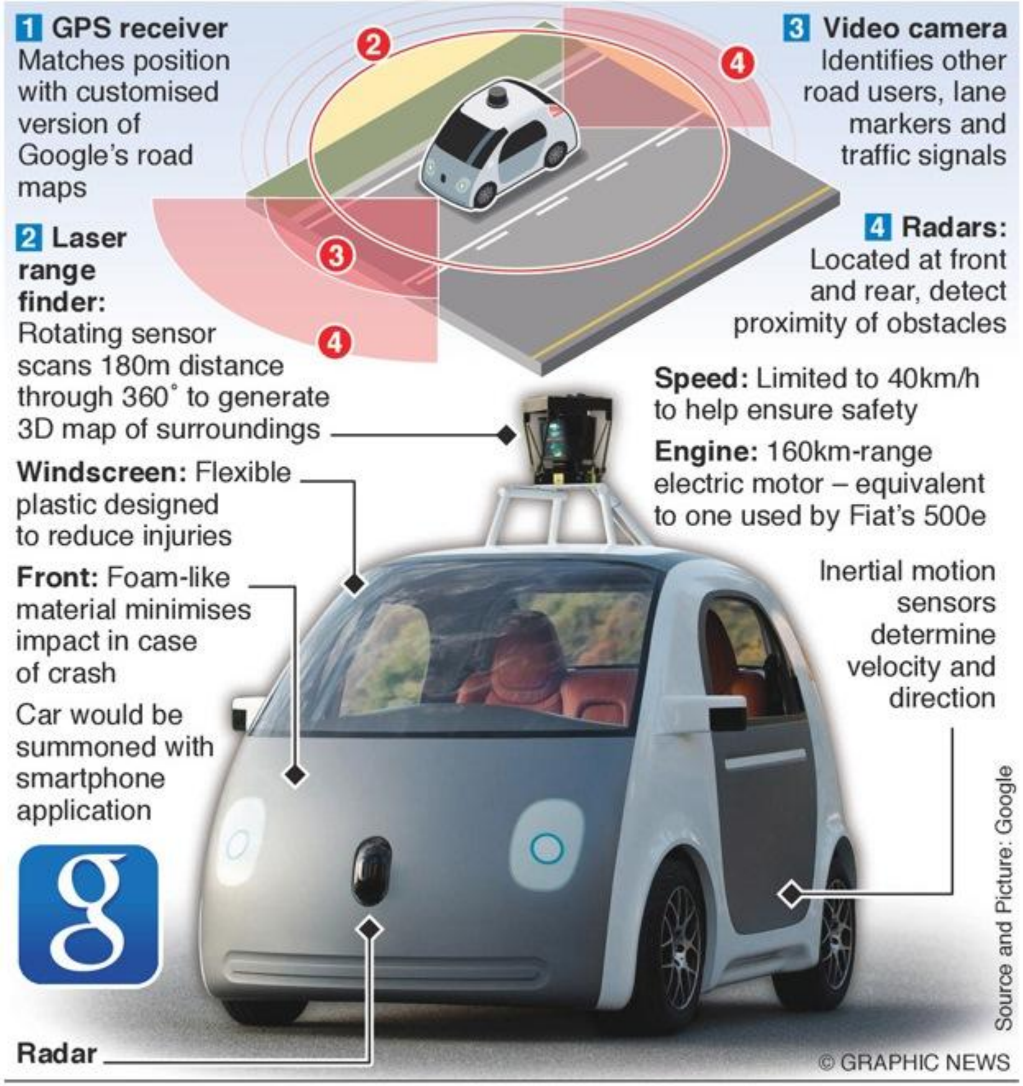


(***Artificial Intelligence: A Modern Approach***, *Stuart Russell and Peter Norvig*)

Sensors

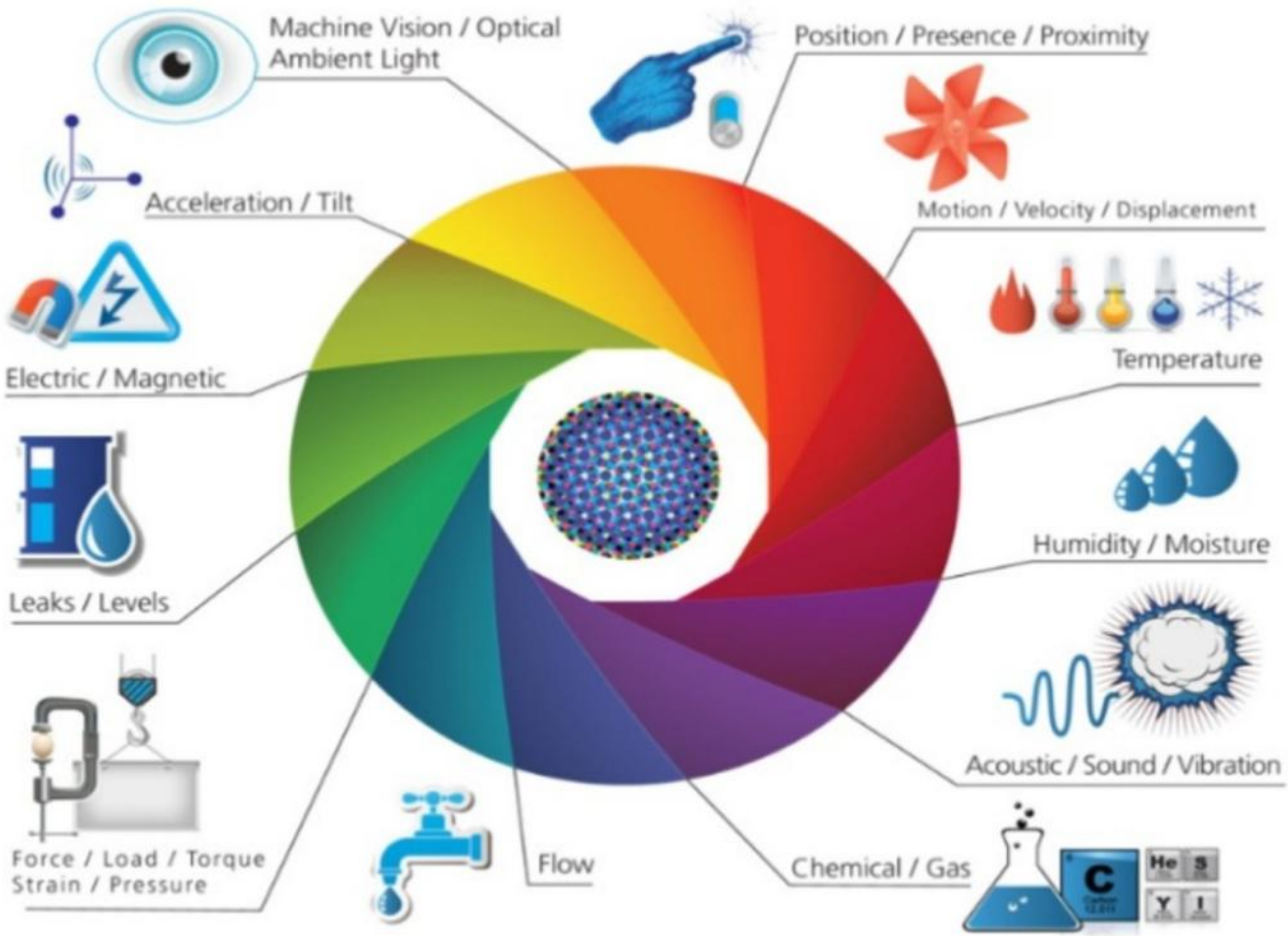


Sensors

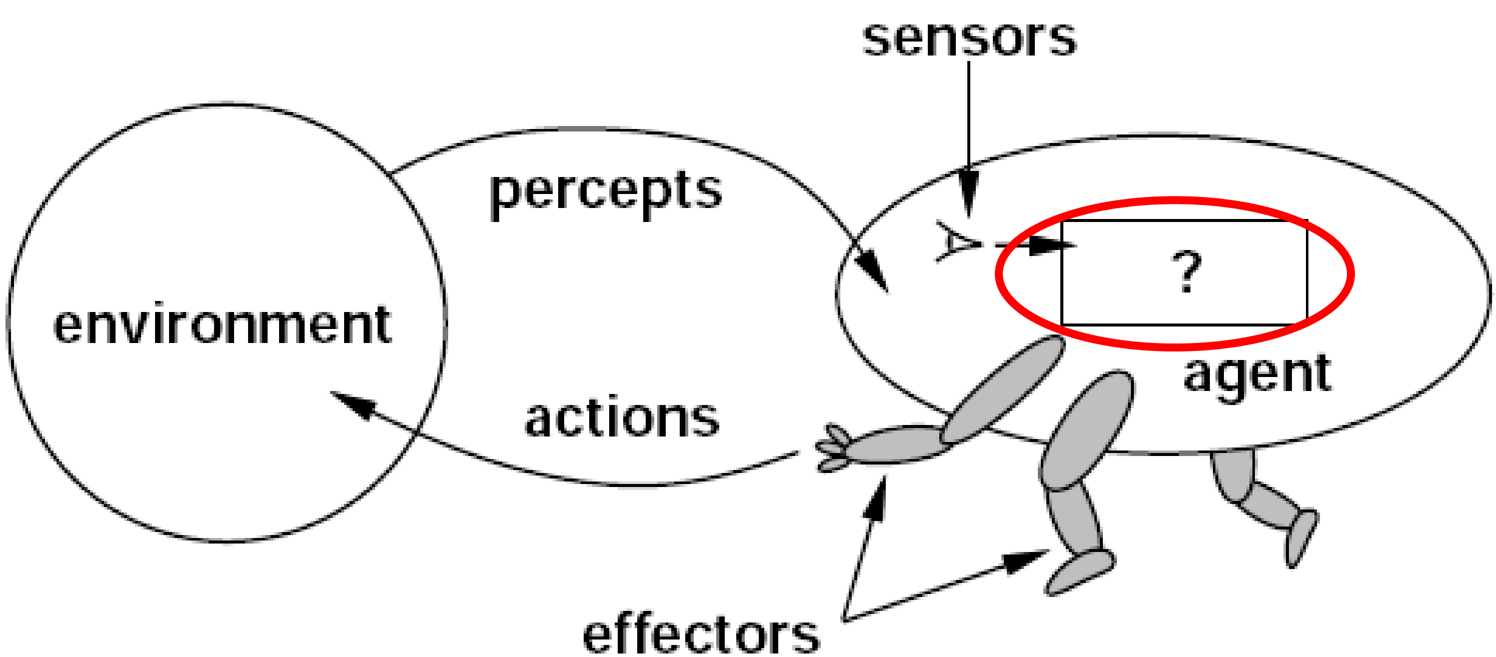


1 GPS receiver
Matches position with customised version of Google's road maps
2 Laser range finder:
Rotating sensor scans 180m distance through 360˚ to generate 3D map of surroundings
3 Video camera
Identifies other road users, lane markers and traffic signals
4 Radars:
Located at front and rear, detect proximity of obstacles
Speed: Limited to 40km/h to help ensure safety
Engine: 160km-range electric motor – equivalent to one used by Fiat's 500e
Windscreen: Flexible plastic designed to reduce injuries
Front: Foam-like material minimises impact in case of crash
Car would be summoned with smartphone application
Inertial motion sensors determine velocity and direction
Radar
Source and Picture: Google
© GRAPHIC NEWS

Trillion Sensor World

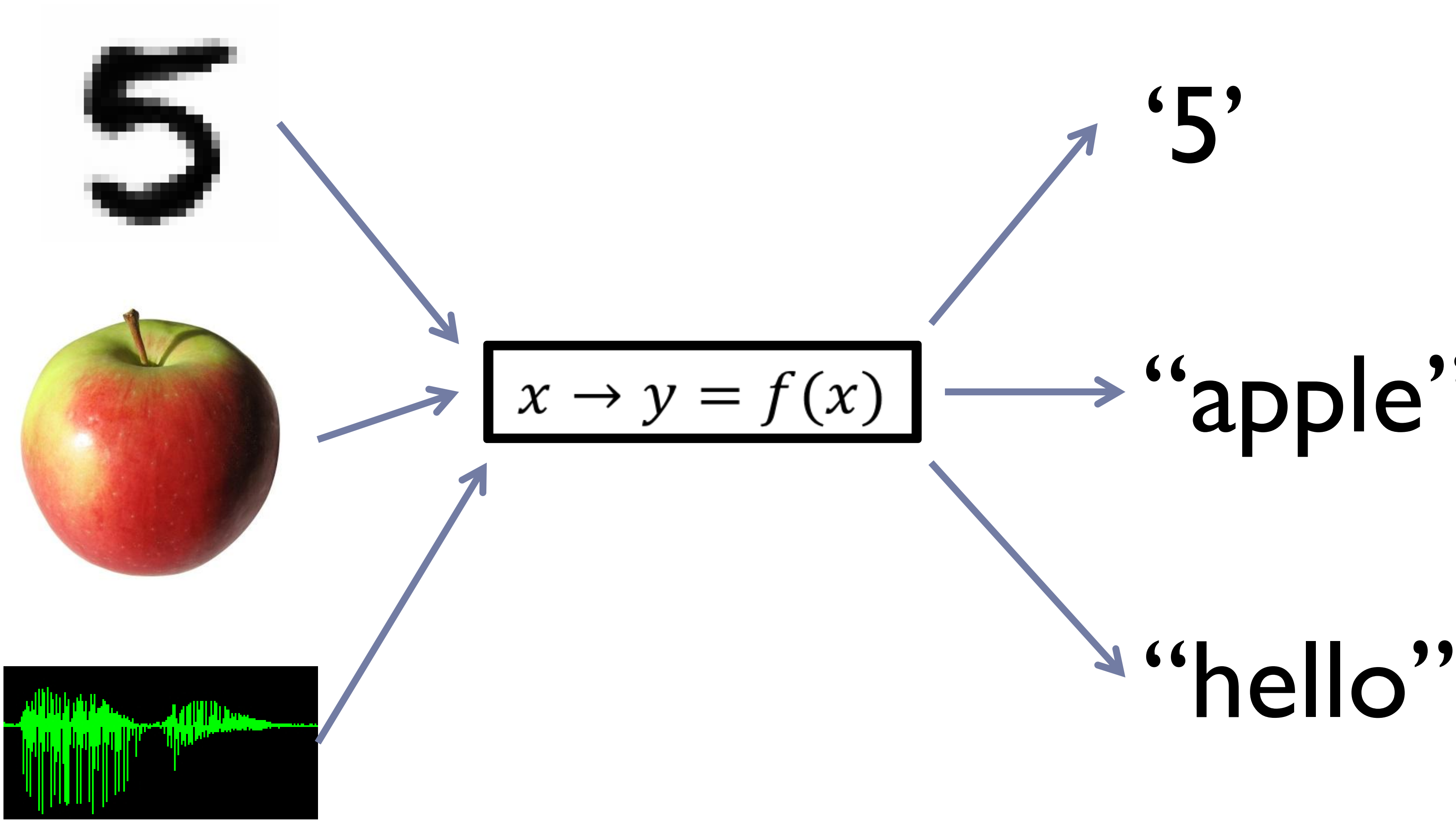


AI and Pattern Recognition



(***Artificial Intelligence: A Modern Approach***, *Stuart Russell and Peter Norvig*)

Pattern Recognition



Why Pattern Recognition is Hard

- **Text detection**

- **Character recognition**

  PLAYA CERRADA

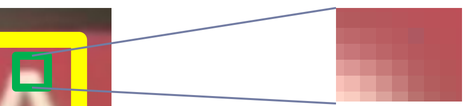
  RECENTE ATAQUE DE TIBURON

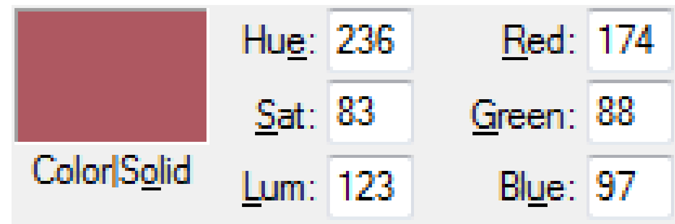
- **Language translation**

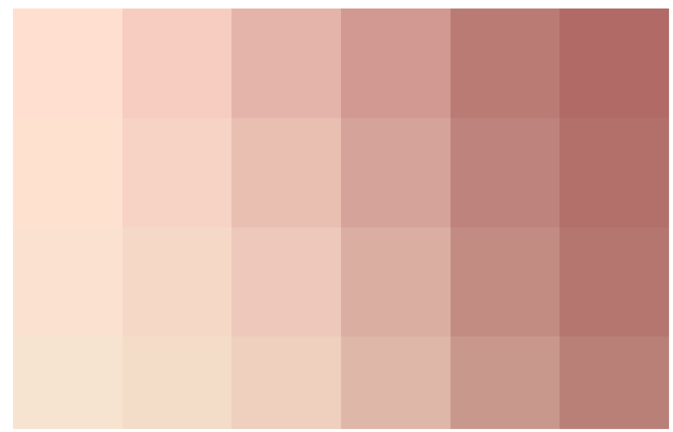
  BEACH CLOSED

  RECENT ATTACK OF SHARK

Why Pattern Recognition is Hard



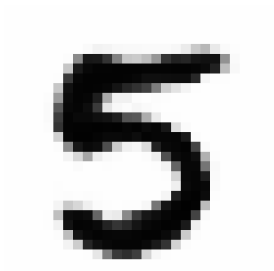




(213,198,170,**174,88,97**,…

Why Pattern Recognition is Hard

PR: Definition

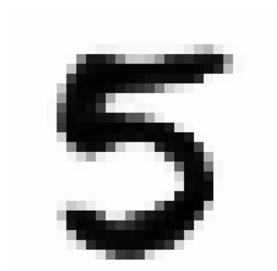


$$\boxed{\begin{aligned} f&: R^d \to Y \\ &x \mapsto y = f(x) \end{aligned}}$$

‘5’

<u>object</u> <u>Pattern Recognition</u> <u>label</u>

Pattern Recognition



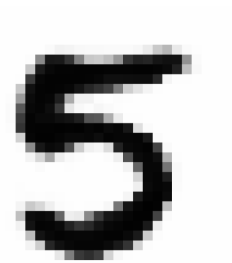
$$\boxed{\begin{aligned} f:\ & R^d \to Y \\ & x \mapsto y = f(x) \end{aligned}}$$

‘5’

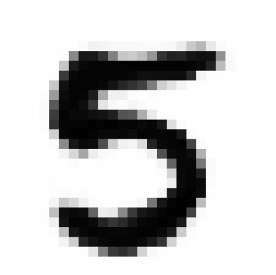
object

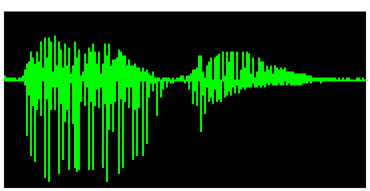
Pattern Recognition

label

 $\in R^?$

Feature Extraction





Feature Extraction

$$\boxed{x \in R^d}$$

$$x \mid \rightarrow y = f(x)$$

$$y = sign(\sum_{i=1}^{d} w_i * x_i + b)$$

<u>classification</u>

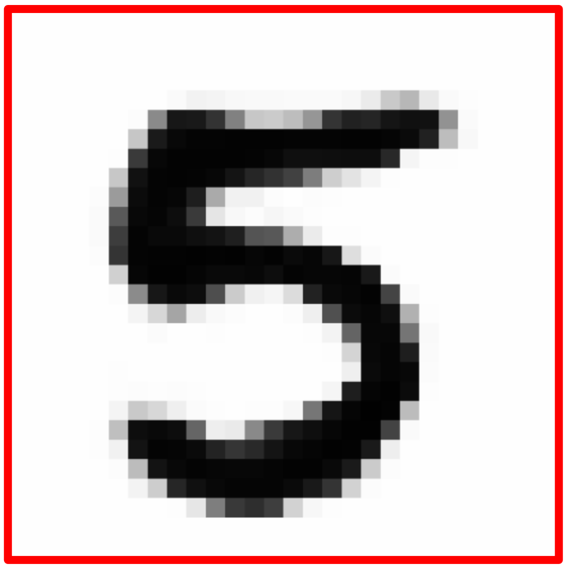
Feature vs. Attribute
(Đặc trưng và thuộc tính)

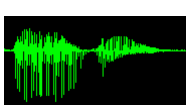
- Attribute
  - Characteristic
  - Quality of a thing
  - Example: weight (kg), volume (cm^3), color (R,G,B)…

- Feature
  - "***Informative***" measurement or characteristics. e.g. improving generalization/prediction performance.
  - Example: Density (kg/m^3)

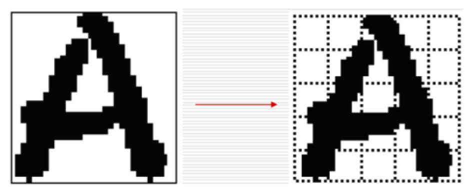
Feature Extraction: ICR

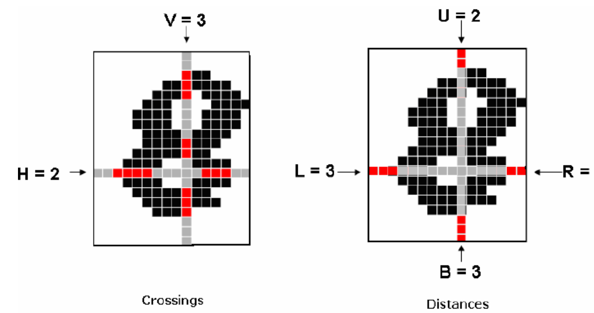
Object





Vector





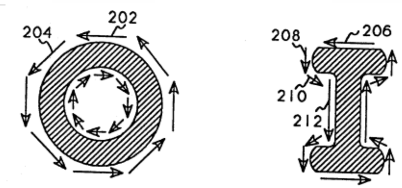


FIG. 2

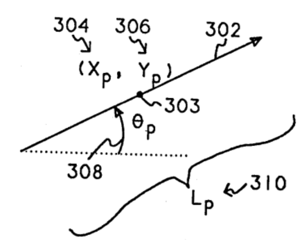
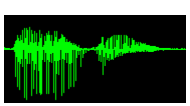
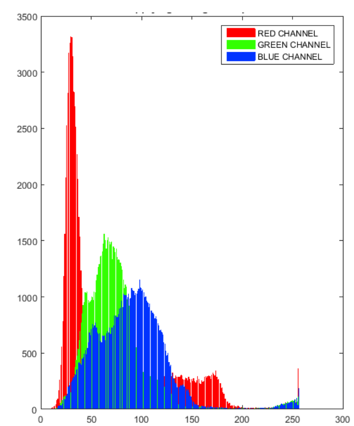


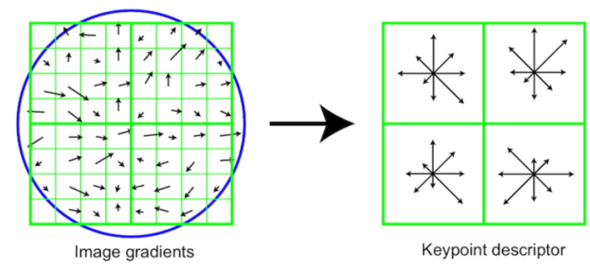
FIG. 3

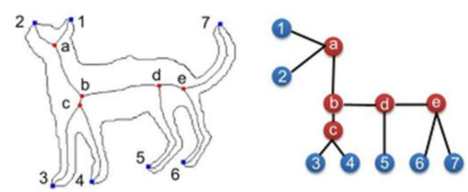
Feature Extraction: Color Image

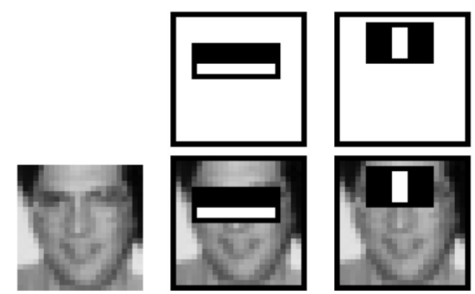
Object **Vector**







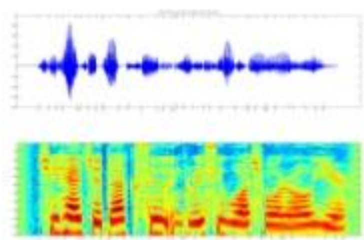




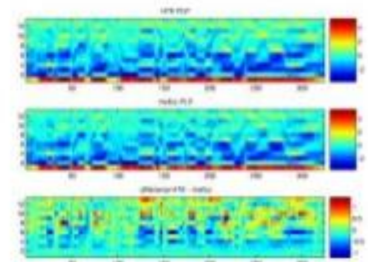
Feature Extraction: Radio Wave

Object

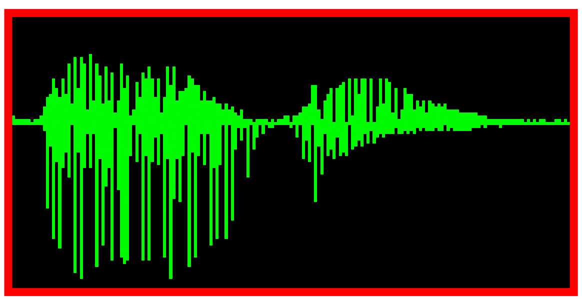
Vector

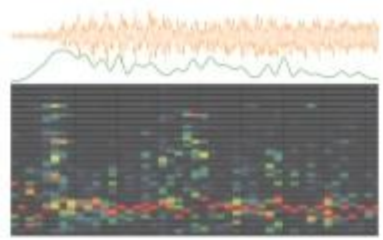


Spectrogram

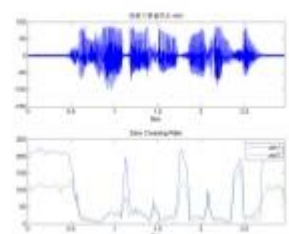


MFCC

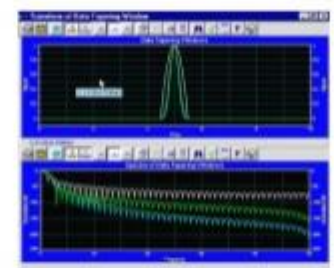




Flux



ZCR



Rolloff

Feature Extraction: Features

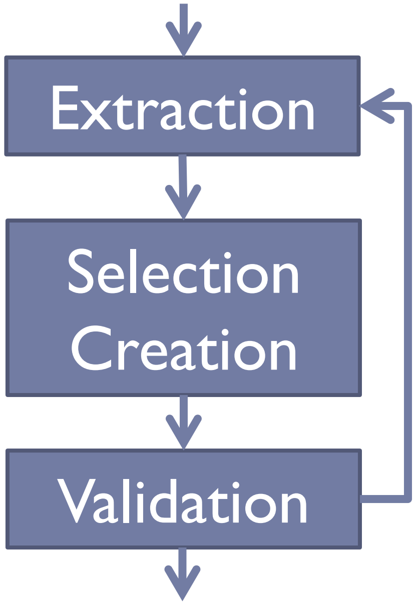
"Coming up with features is difficult, time-consuming, requires expert knowledge." (Andrew Ng, Machine Learning and AI via Brain simulations)

- Informative
  - Help improving performance
- Non-redundant
  - Removed without performance degradation
- Explainable
  - Understandable by human
- …

Feature: Engineering vs. Learning

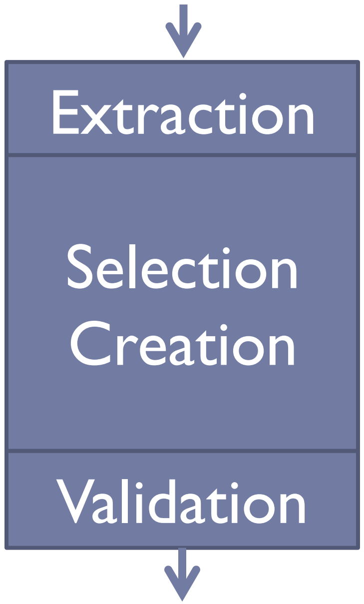
Feature Engineering

- ***Using domain knowledge*** to create features that make machine learning algorithms work.

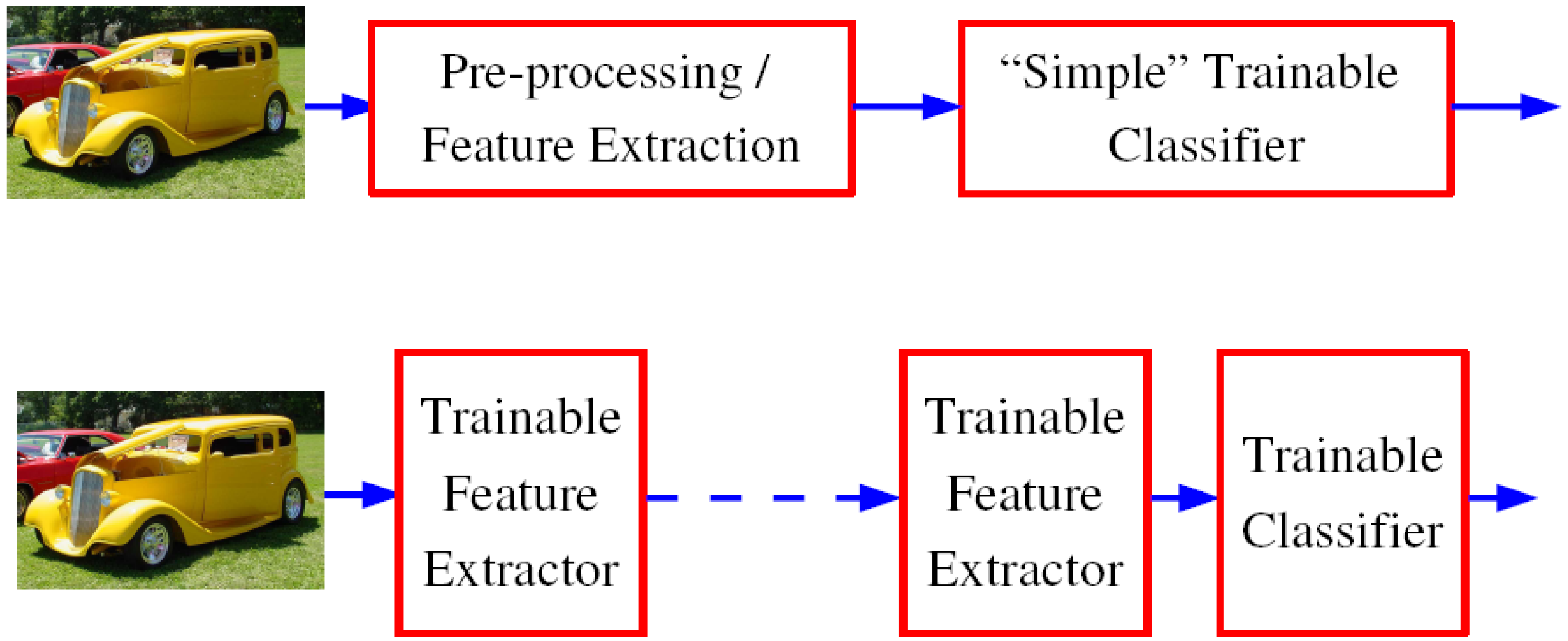


Feature Learning

- ***Automatically*** create features that make machine learning algorithms work.

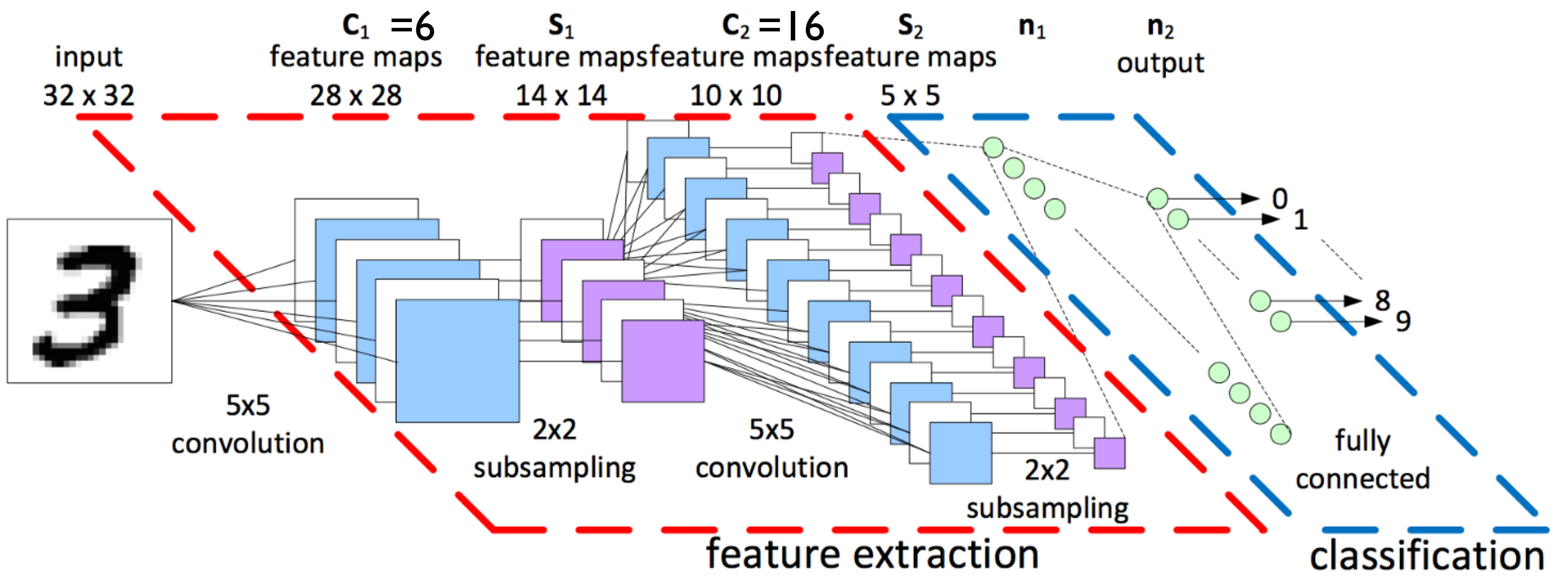


Feature: Engineering vs. Learning



(Yann LeCun, 2010)

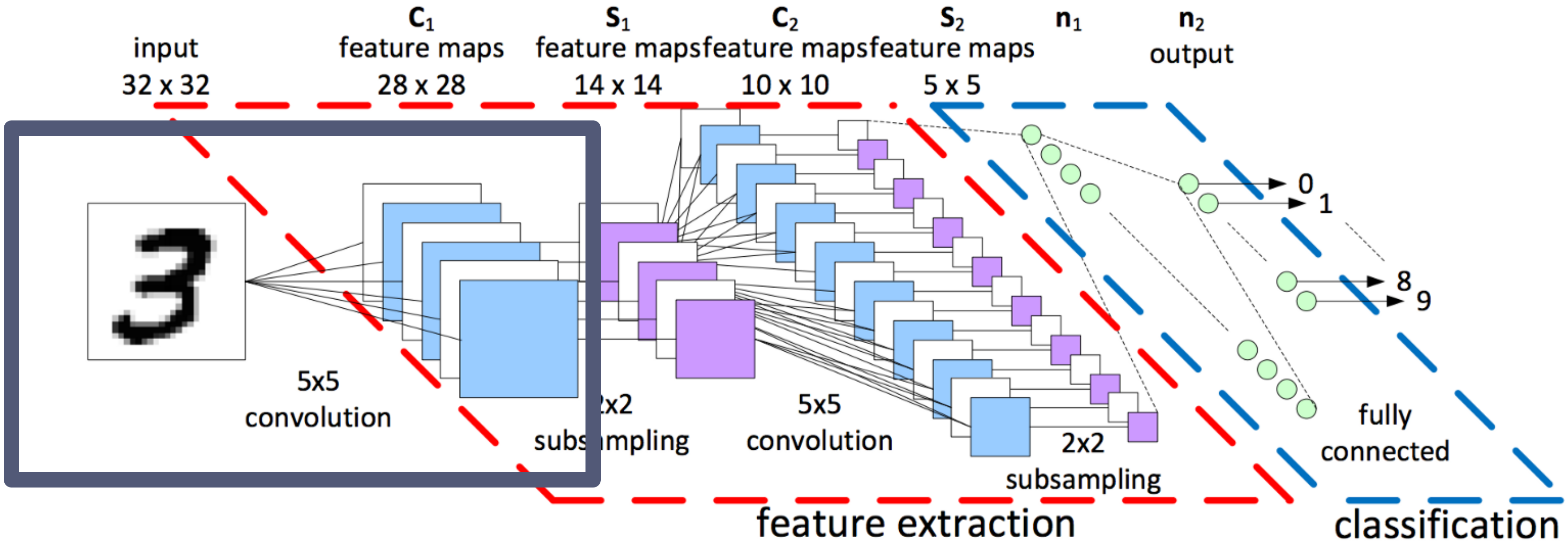
Handwritten Digit Recognition: LeNet-5

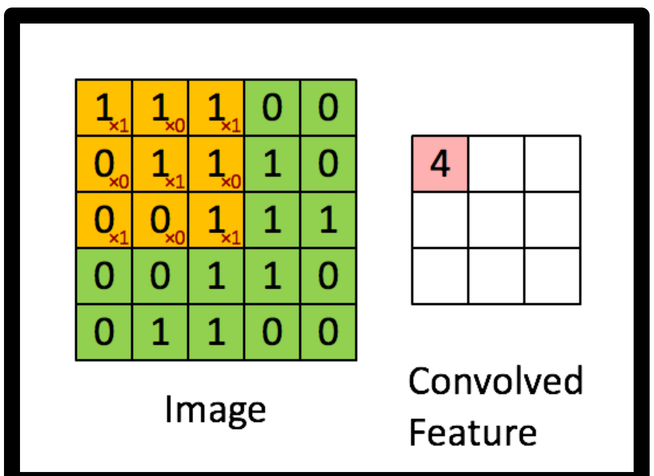


MNIST Error Rates

k-NN	1-layer NN	2-layer NN	SVM	LeNet-4	LeNet-5
5.0	12.0	4.7	1.4	1.1	**0.95**

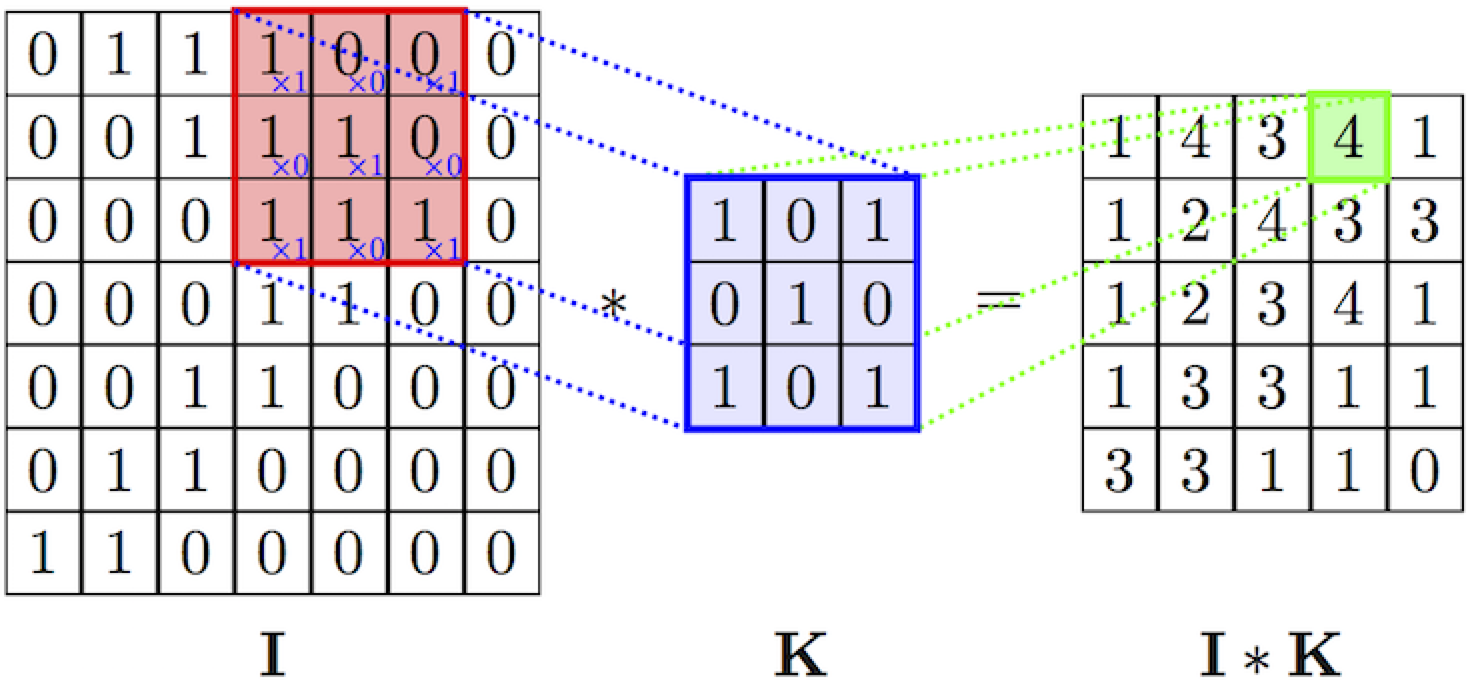
Convolution Process



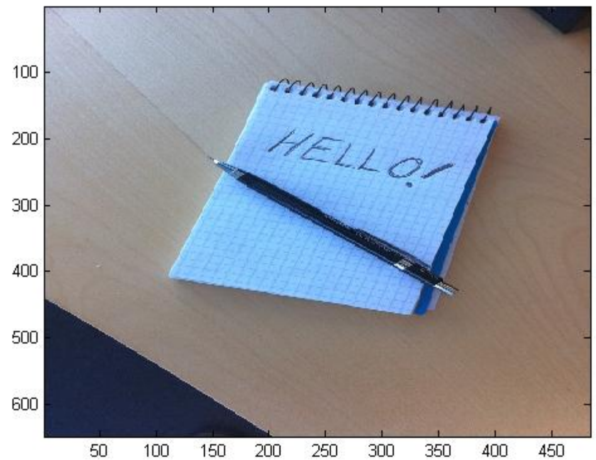


Convolution Operator

$$(I * K)_{xy} = \sum_{i=1}^{h} \sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y+j-1}$$

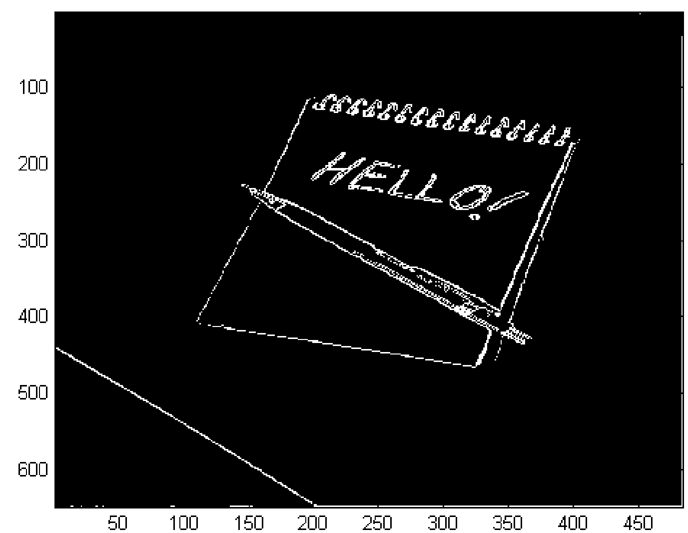


Edge Detection Filter / Kernel

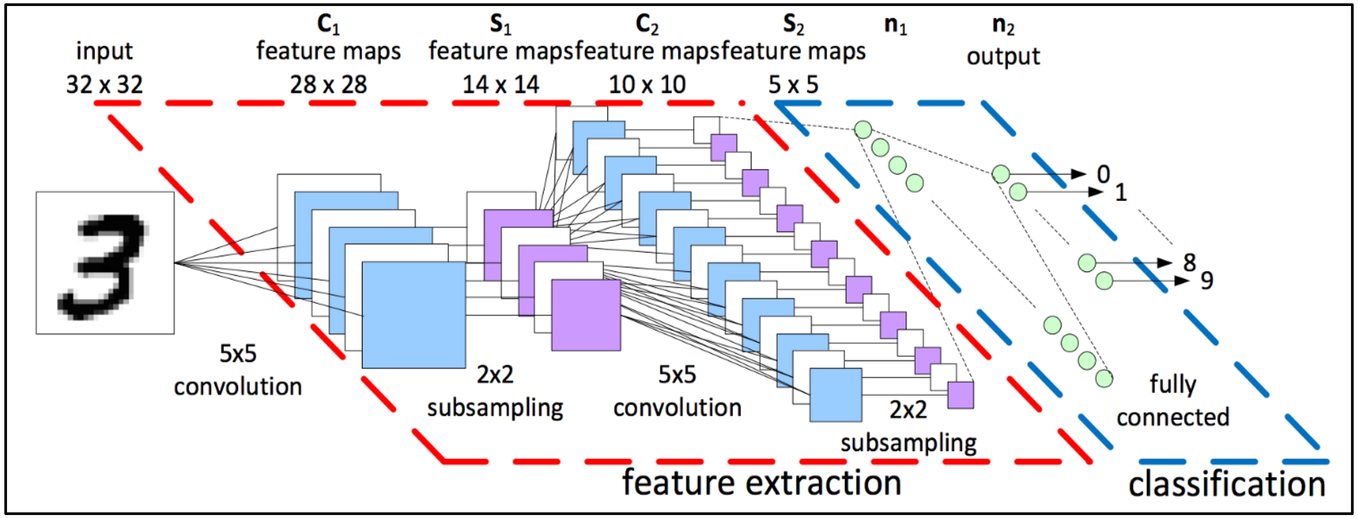


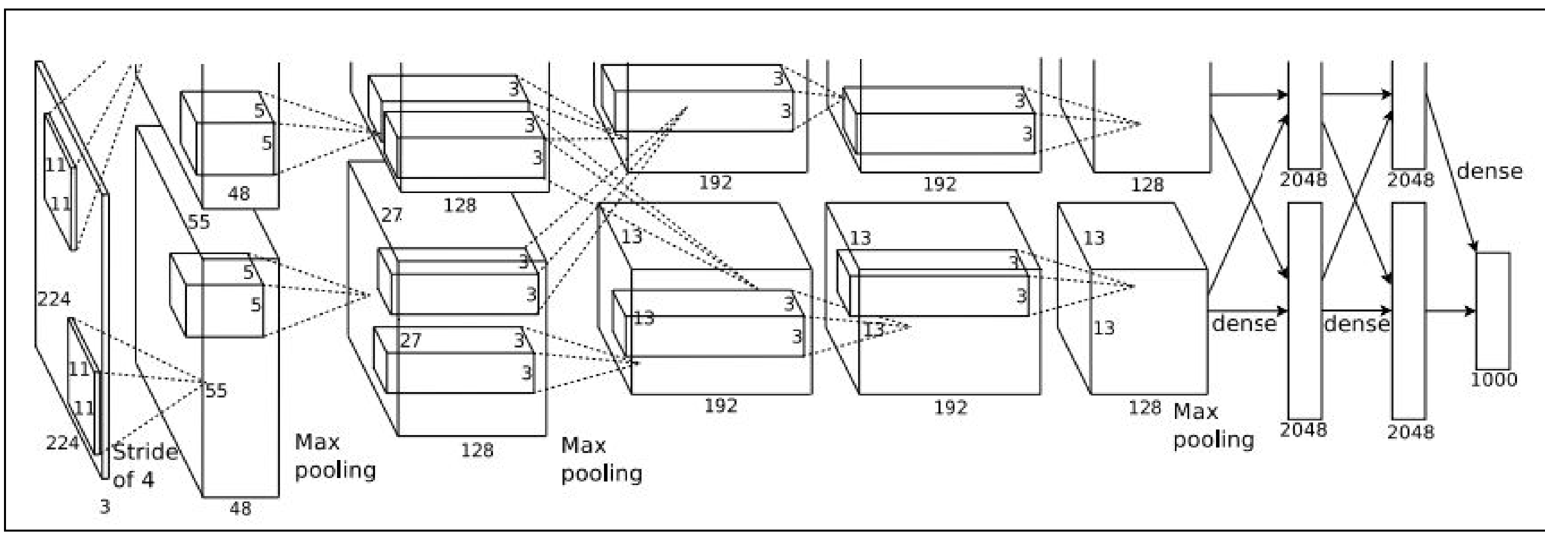
-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

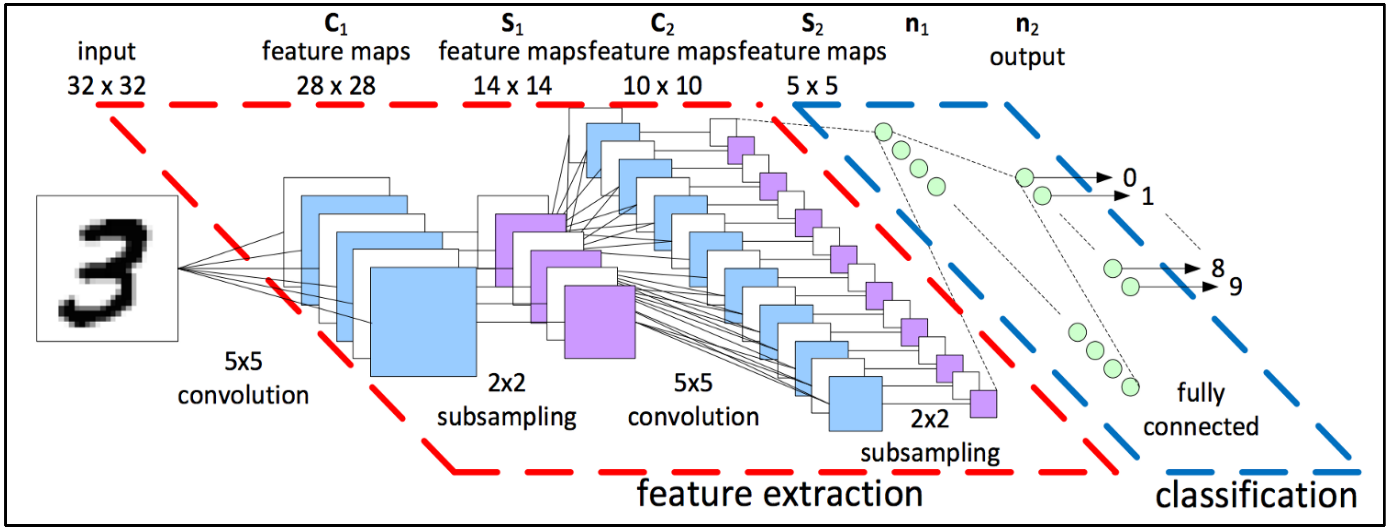


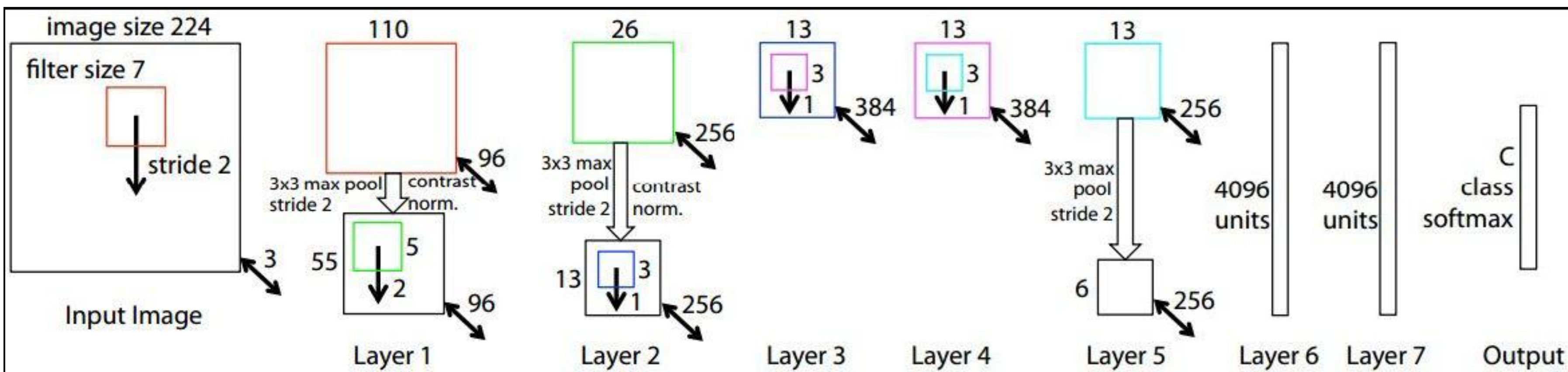
LeNet-5, AlexNet



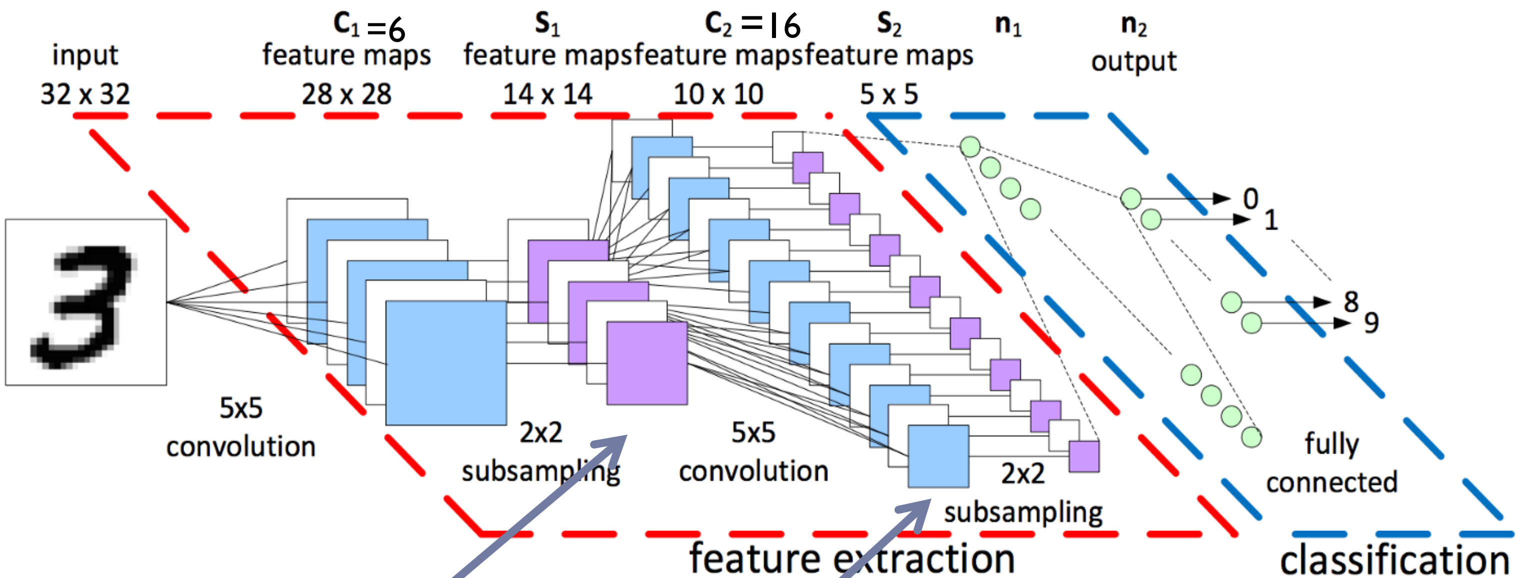


LeNet-5, VGGNet





LeNet-5: “Handcrafted” Convolution



C_2

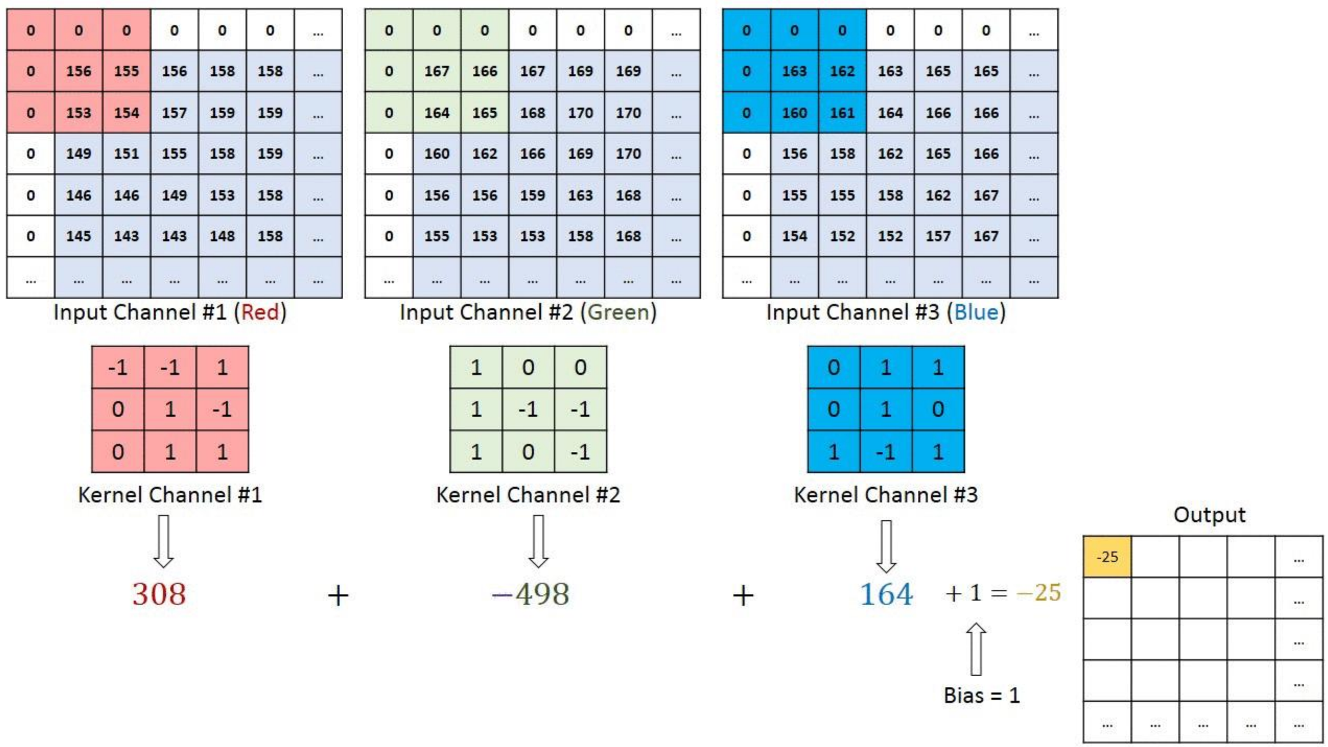
S_1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

"Normal" Convolution

$$Y_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{i,j}^{(l)} * Y_j^{(l-1)}$$

$$\boxed{m_1{}^{(l-1)} = 3}$$



LeNet-5: “Handcrafted” vs. “Normal” Convolution

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

1,516

parameters

$$Y_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{i,j}^{(l)} * Y_j^{(l-1)}$$

?

parameters

$m_1{}^{(l-1)} = 6, m_1{}^{l} = 16, K = 5x5.$

LeNet-5: “Handcrafted” vs. “Normal” Convolution

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

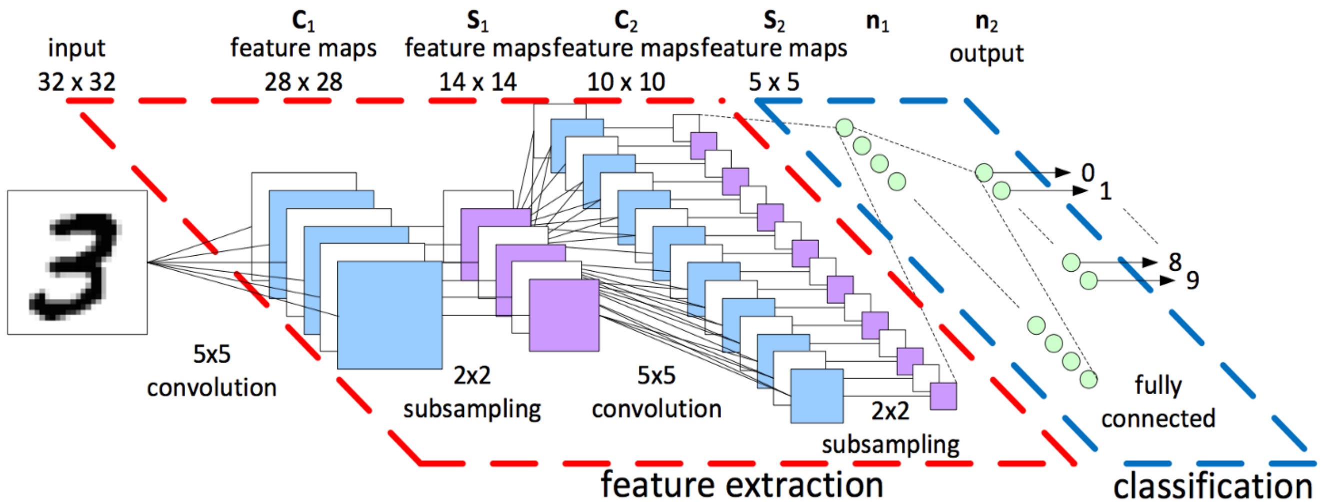
1,516
parameters

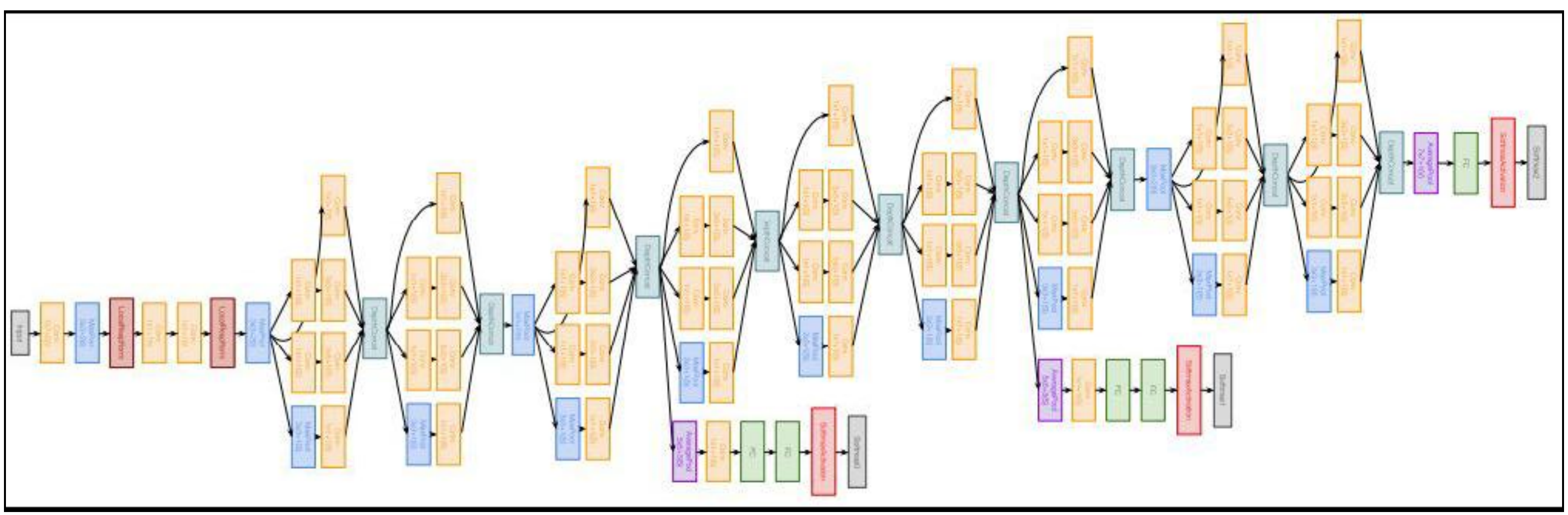
$$Y_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{i,j}^{(l)} * Y_j^{(l-1)}$$

$\boldsymbol{m_1^{(l-1)} = 6, m_1^{\,l} = 16, K = 5x5.}$

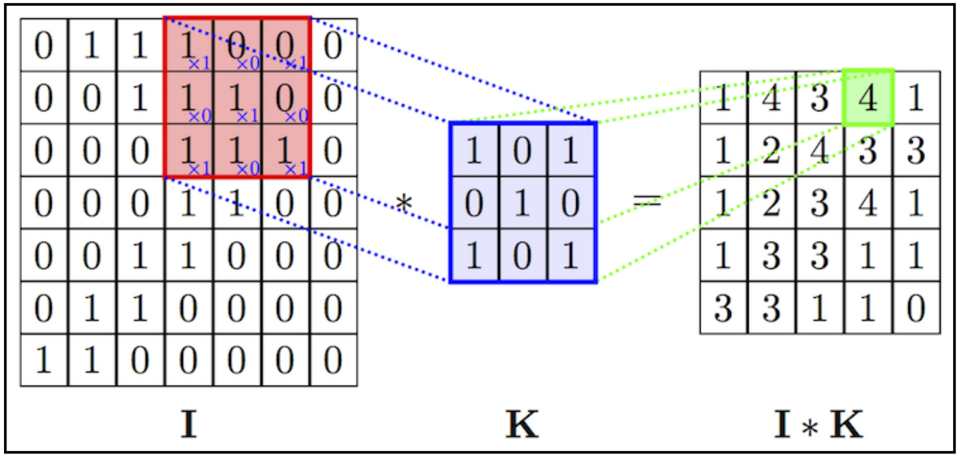
5x5x6x16+
(2.400+) parameters

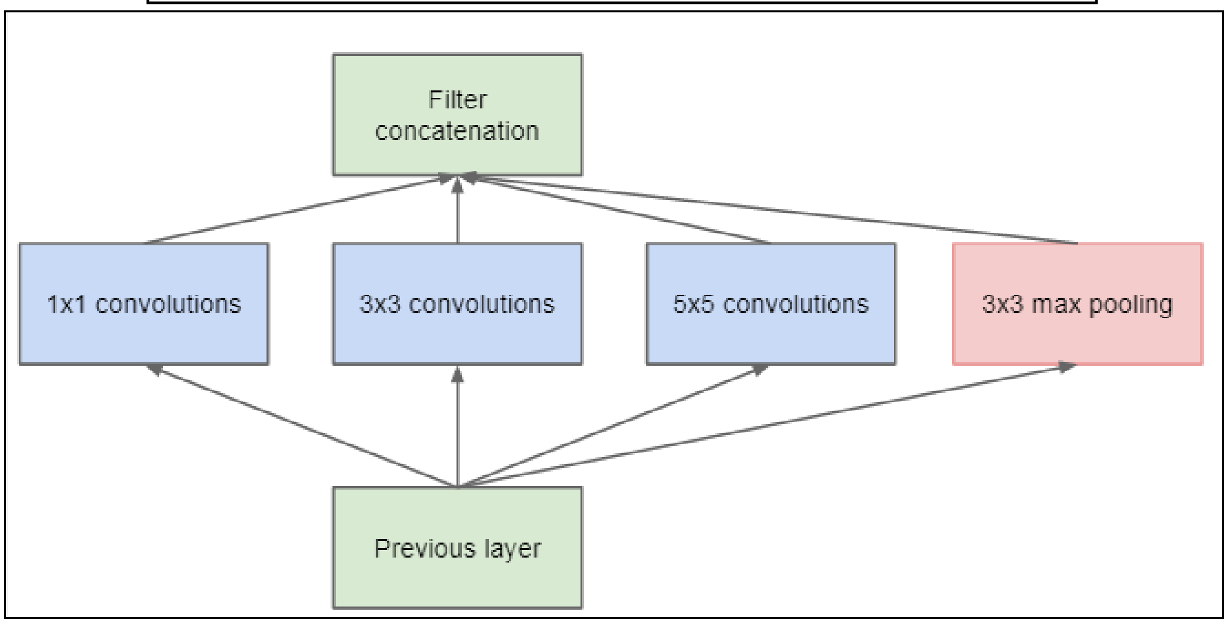
LeNet-5, GoogLeNet



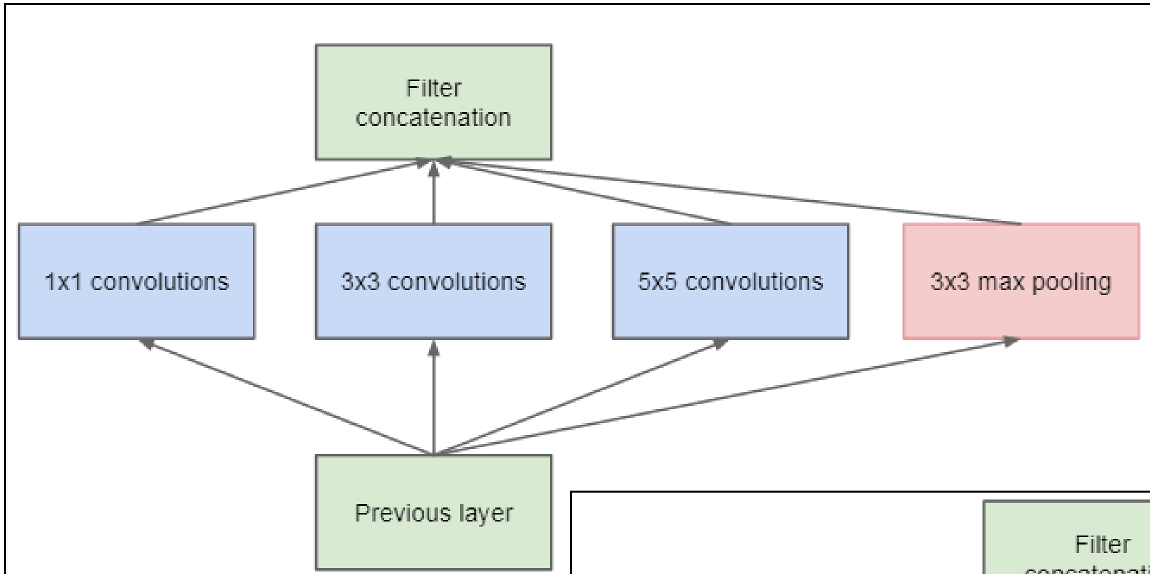


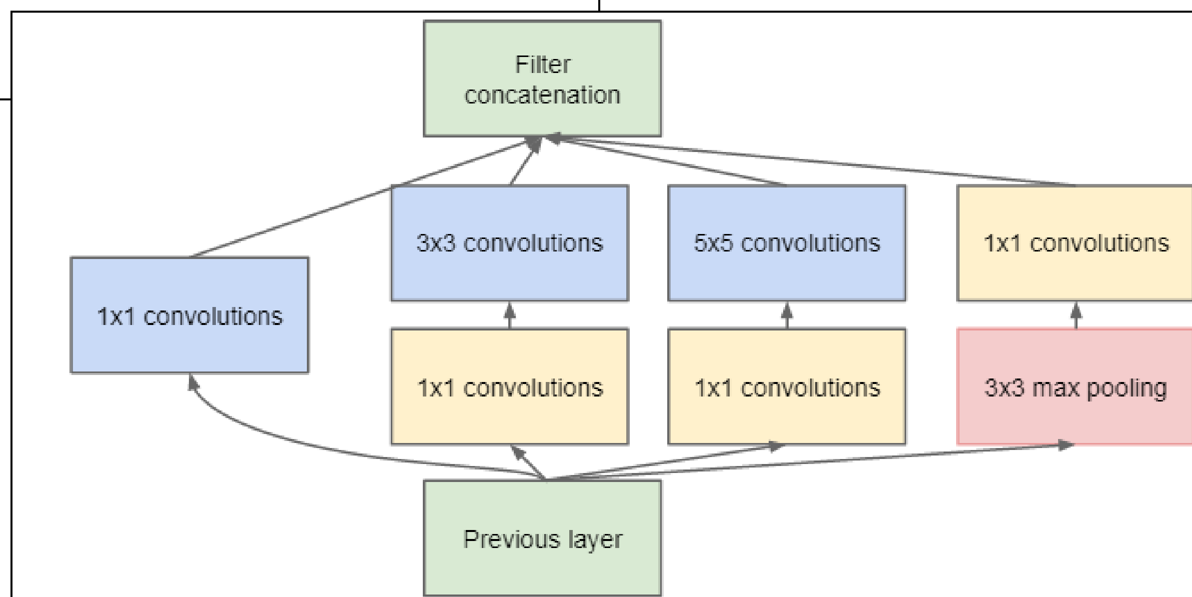
Convolution, Reception



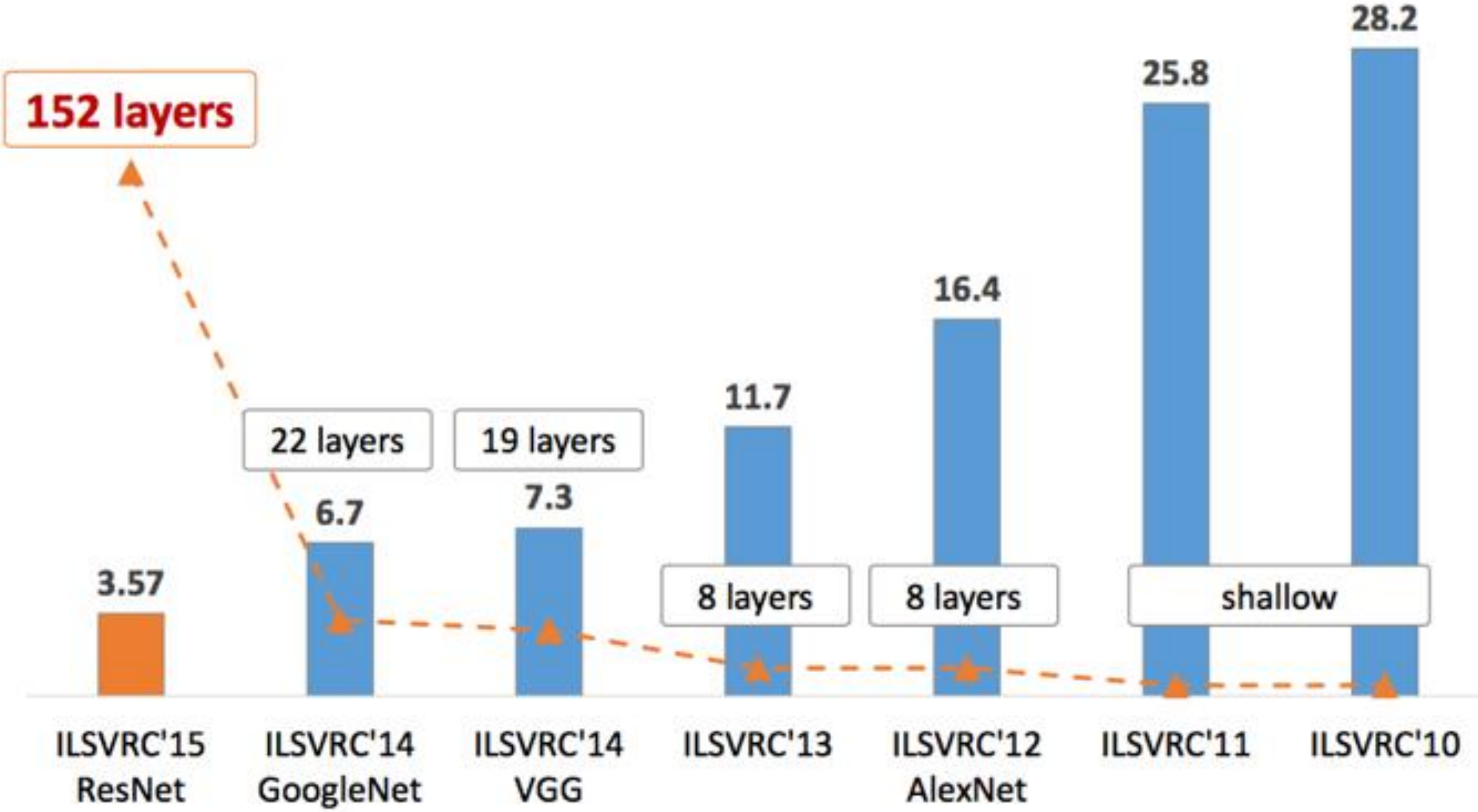


Reception,
Reception with Dimension Reduction



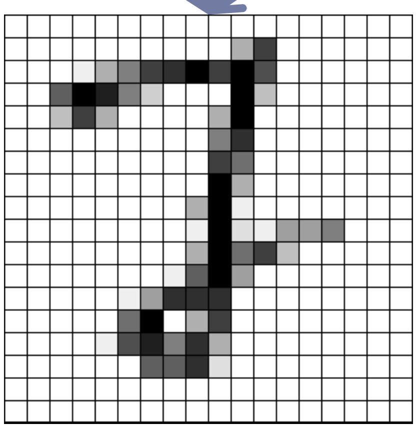


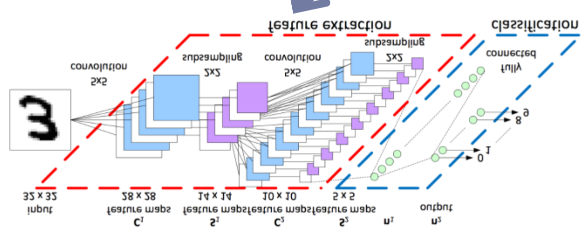
#Layers vs. Performance

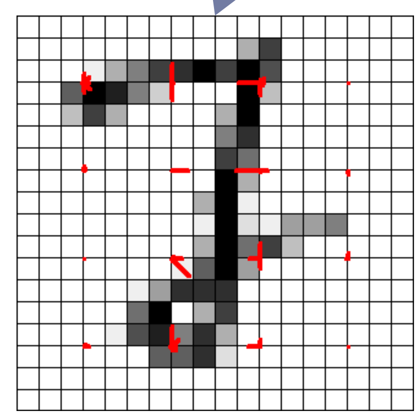


MNIST Revisited

k-NN	2-layer NN	SVM RAW	LeNet-5	MCDNN	SVM HOG
5.0	4.7	1.4	0.95	**0.23**	0.61







Gradient Feature

- Filter mask

-1	0	1
-2	0	2
-1	0	1

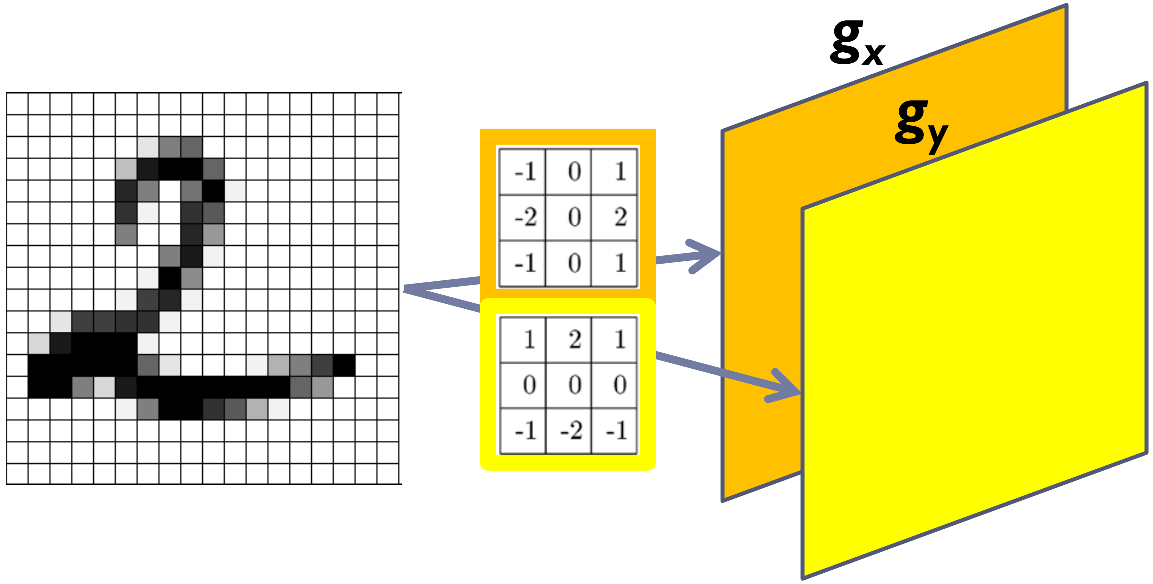
1	2	1
0	0	0
-1	-2	-1

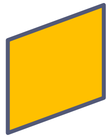
- Feature $\mathbf{g}(x, y) = [g_x, g_y]^{\mathrm{T}}$

$$
\begin{aligned}
g_x(x, y) = {} & f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1) \\
& - f(x-1, y-1) - 2f(x-1, y) \\
& - f(x-1, y+1),
\end{aligned}
$$

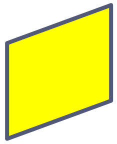
$$
\begin{aligned}
g_y(x, y) = {} & f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1) \\
& - f(x-1, y-1) - 2f(x, y-1) \\
& - f(x+1, y-1).
\end{aligned}
$$

Gradient Feature: $[g_x, g_y]$



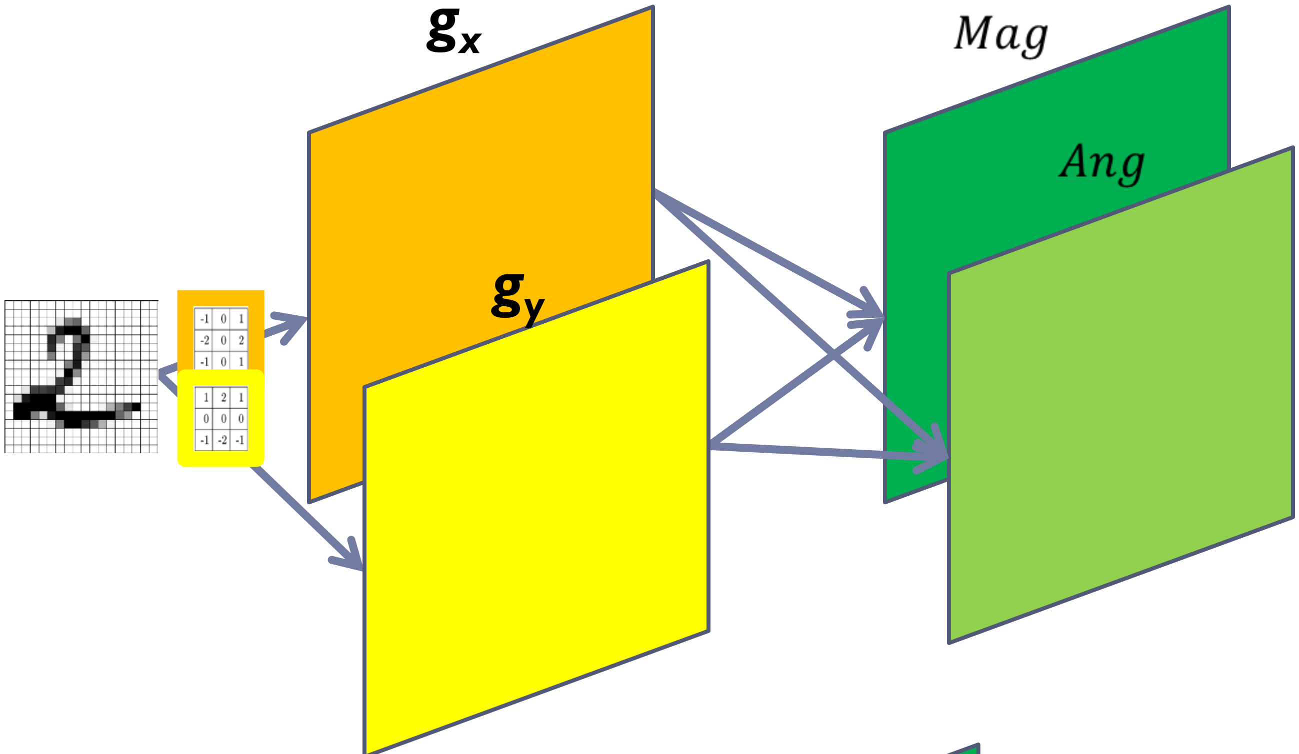


$$\begin{aligned} g_x(x,y) = {} & f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1) \\ & - f(x-1,y-1) - 2f(x-1,y) \\ & - f(x-1,y+1), \end{aligned}$$



$$\begin{aligned} g_y(x,y) = {} & f(x-1,y+1) + 2f(x,y+1) + f(x+1,y+1) \\ & - f(x-1,y-1) - 2f(x,y-1) \\ & - f(x+1,y-1). \end{aligned}$$

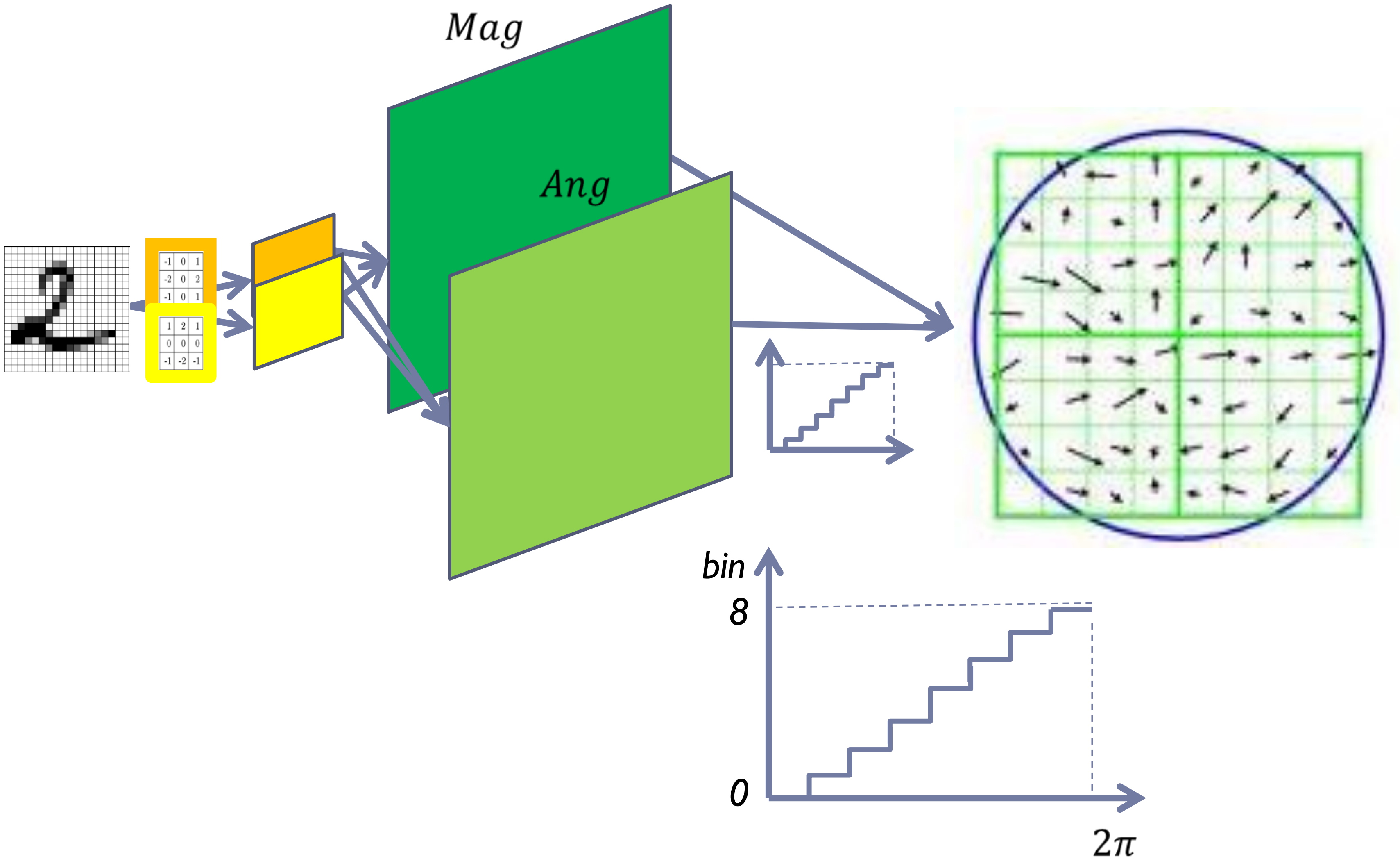
Gradient Feature: Magnitude and Angle



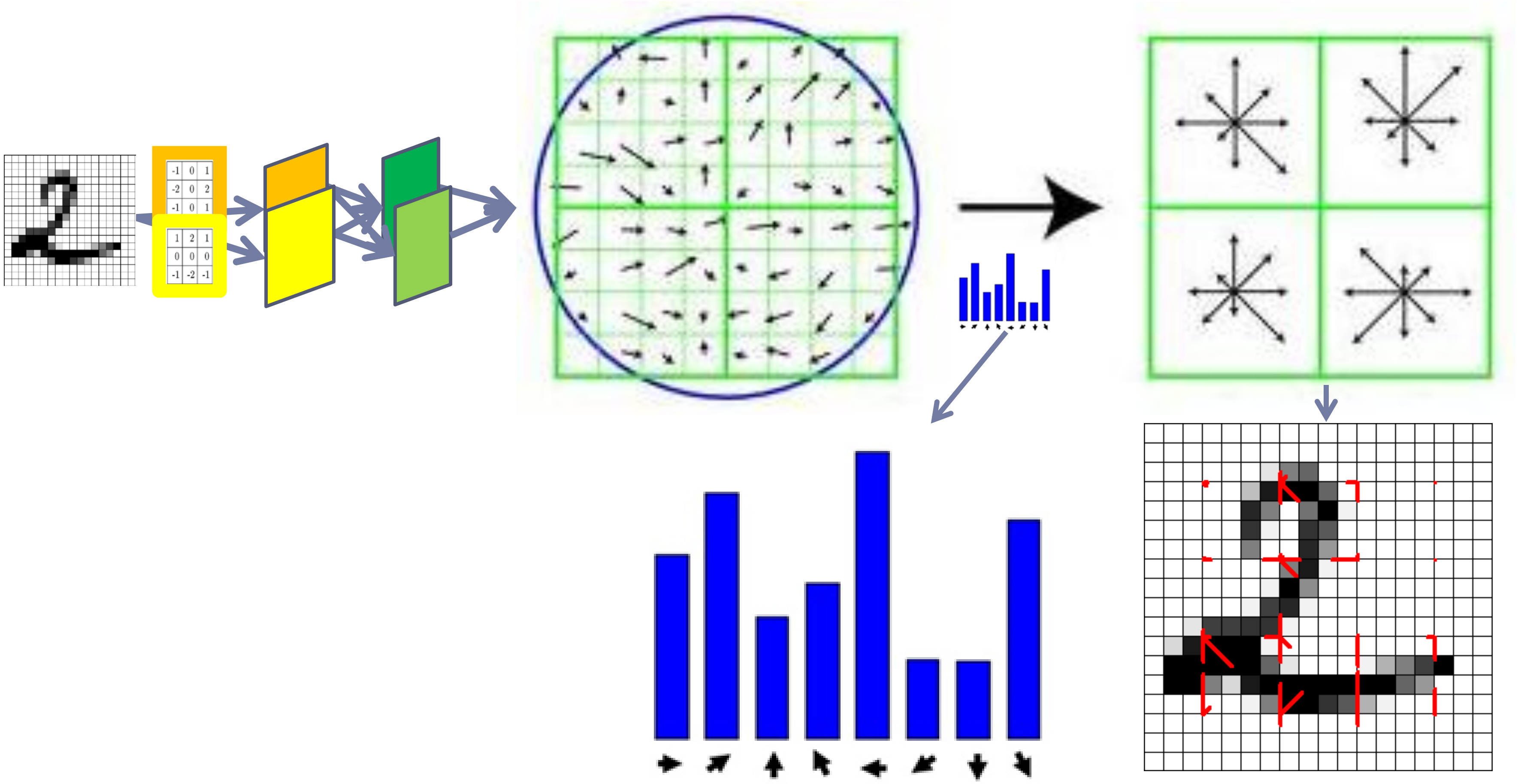
$$Mag = \sqrt{{g_x}^2 + {g_y}^2}$$

$$Ang = arctan\frac{{g_x}^2}{{g_y}^2}$$

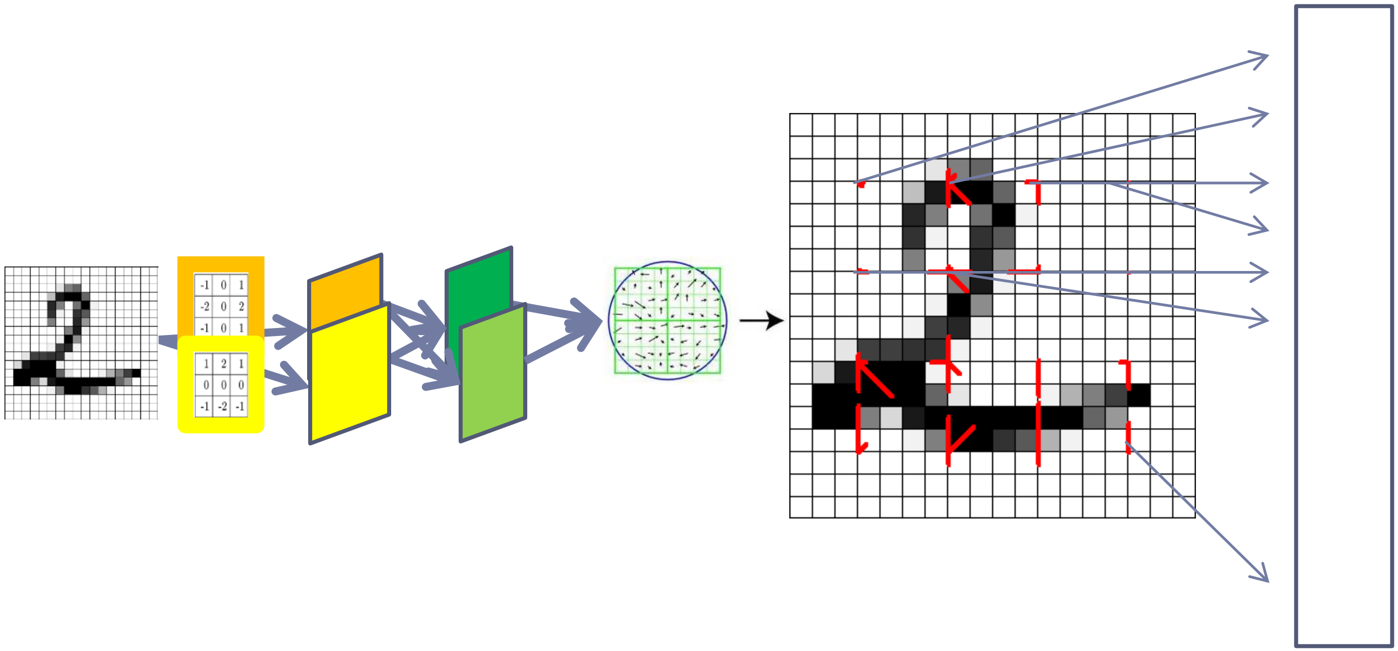
Gradient Feature: Discrete Direction



Discrete Direction: (Sum) Sampling



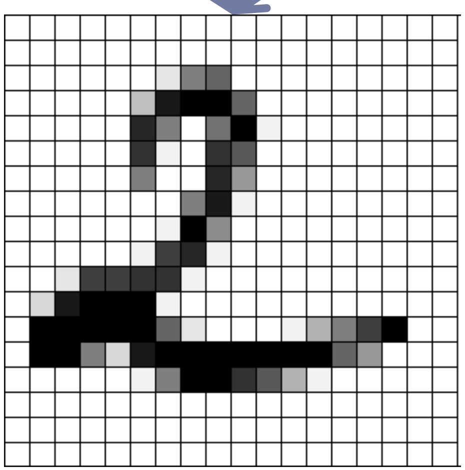
Discrete Direction: Concatenation

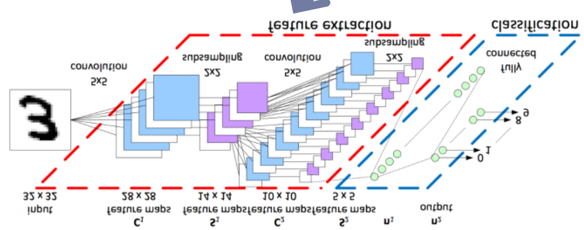


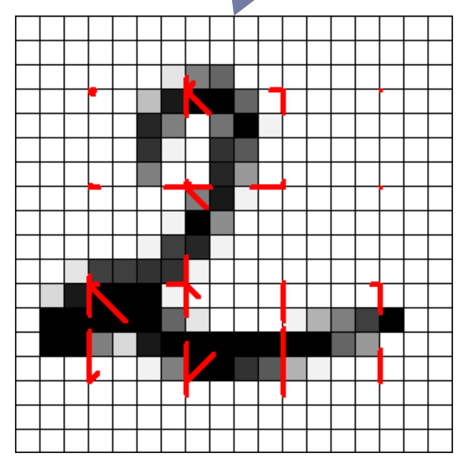
4x4x8 dimensions

MNIST Test Error Rate

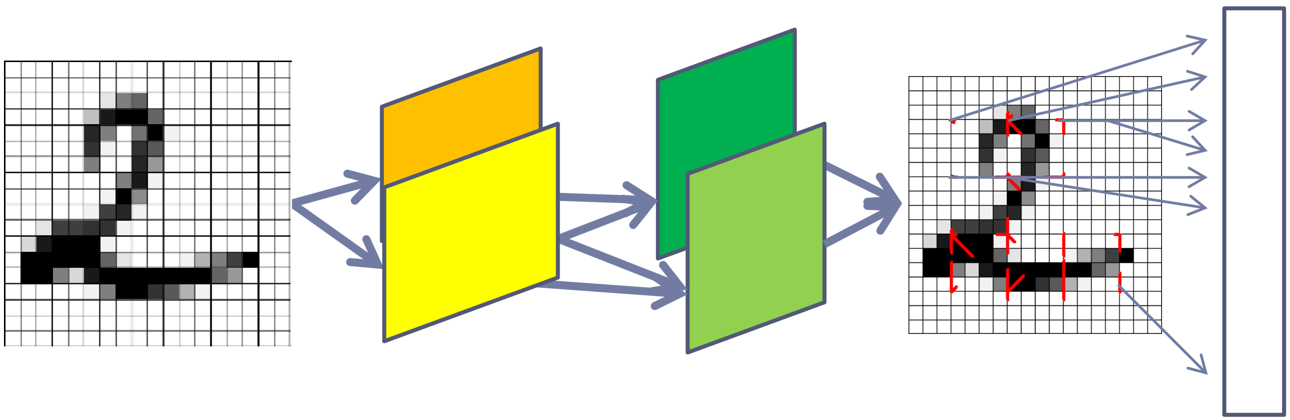
k-NN	2-layer NN	SVM RAW	LeNet-5	Mul.Col. DNN	SVM HOG
5.0	4.7	1.4	0.95	**0.23**	0.61

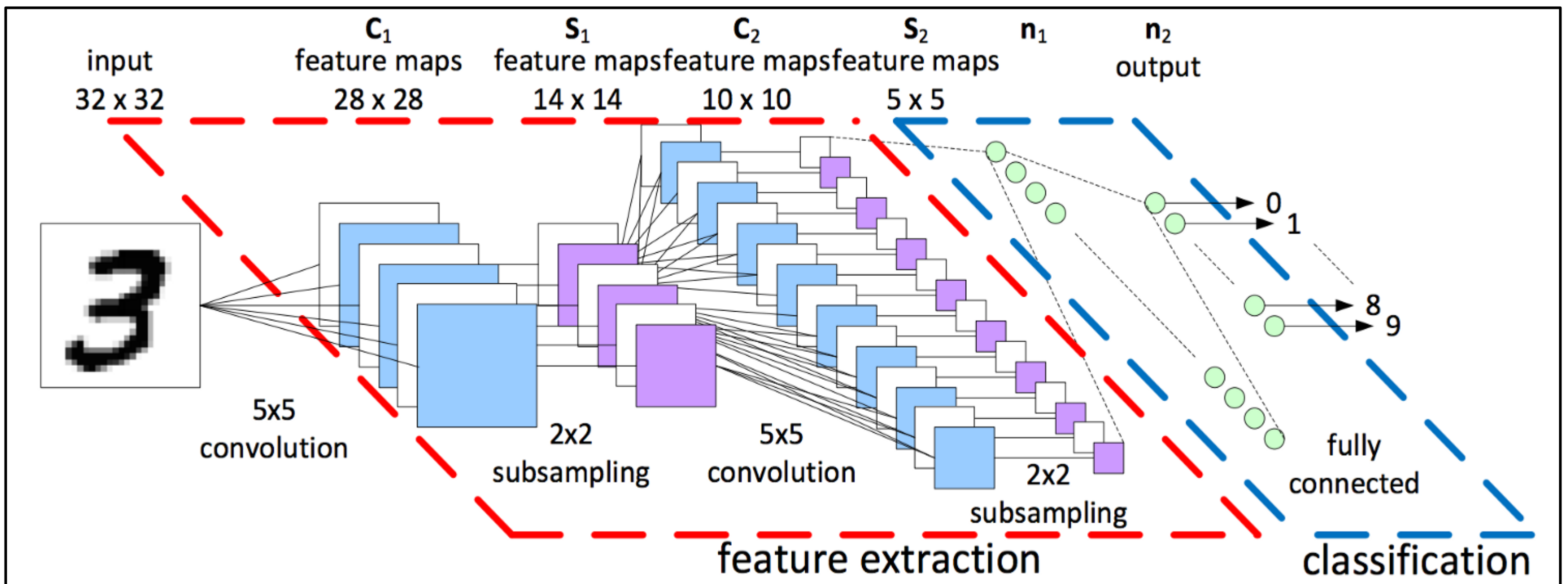






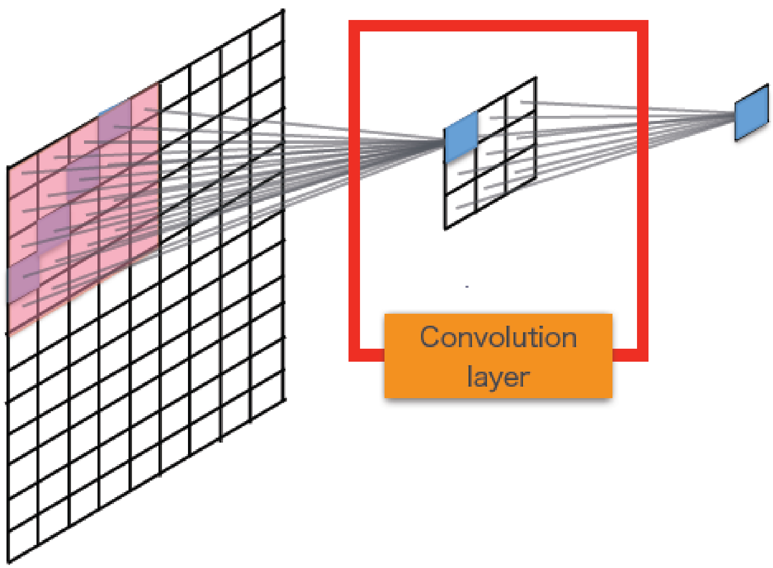
HOG vs. LeNet-5





CNN Convolution vs. Filter

$$(I * K)_{xy} = \sum_{i=1}^{h} \sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y+j-1}$$



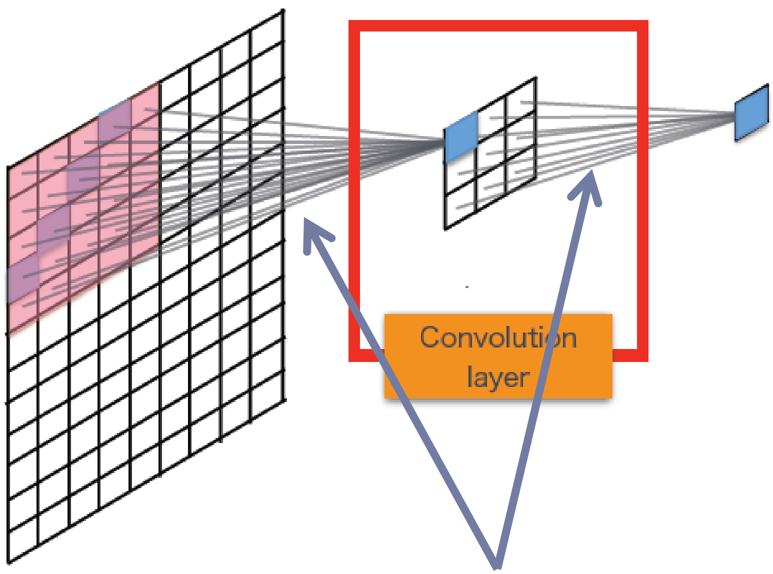
-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

<u>**Convolution**</u> <u>**Filter**</u>

CNN Convolution vs. Filter

$$(I * K)_{xy} = \sum_{i=1}^{h} \sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y+j-1}$$

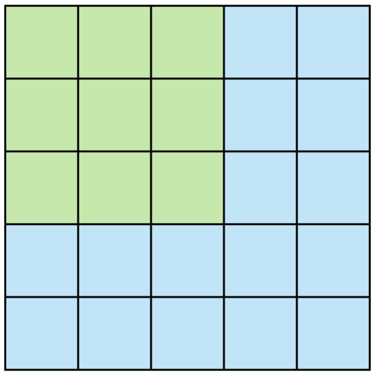


$$\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \qquad \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$

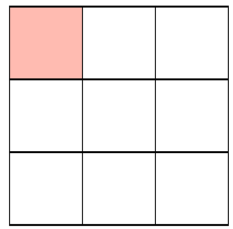
Deep, trainable

Shallow, handcrafted

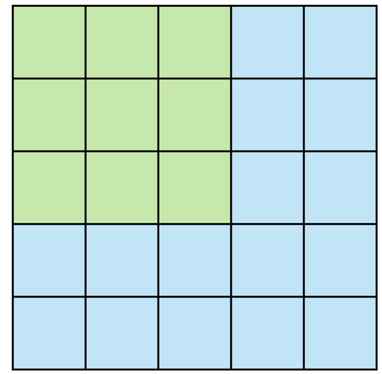
Stride and Padding



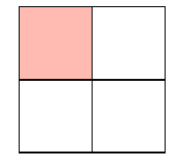
Stride 1



Feature Map

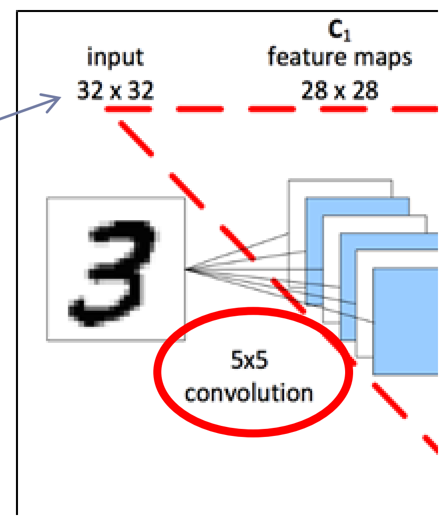


Stride 2

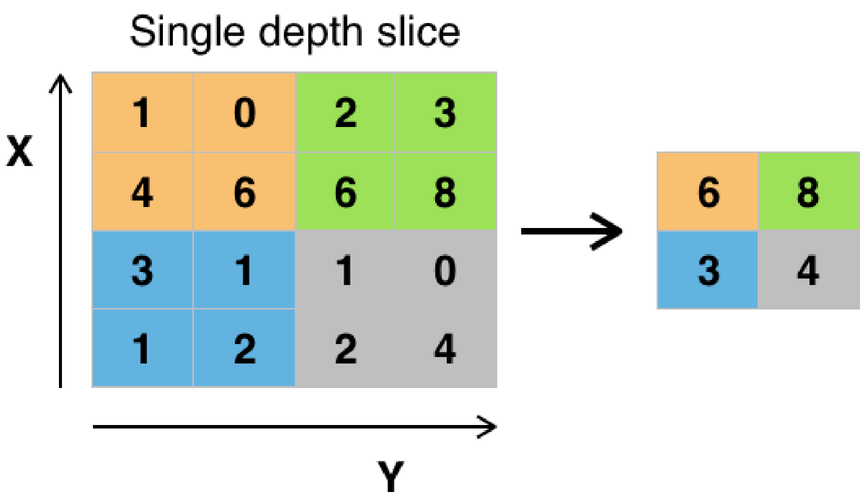


Feature Map

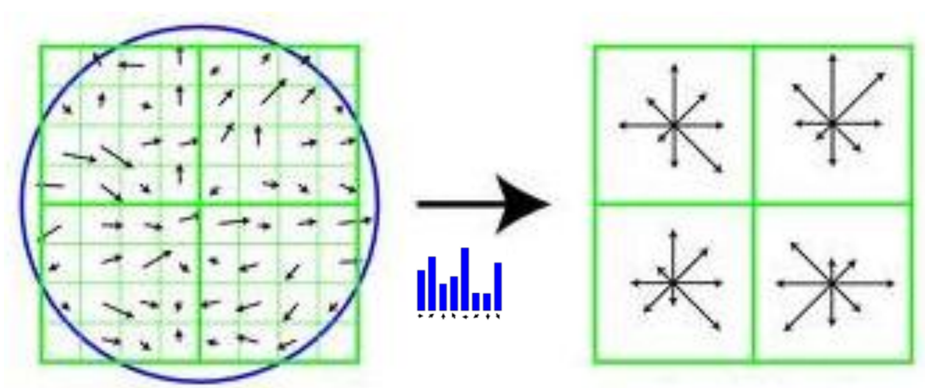
28x28



Pooling/Sampling



Example of Maxpool with a 2x2 filter and a stride of 2



Objective:

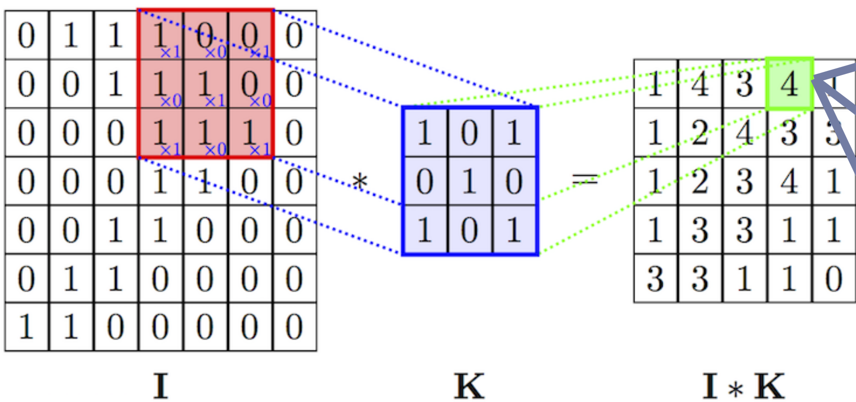
- Improve space-invariance
- Reduce parameters
- More abstract features

Methods:

- Max pooling
- Sum/Mean pooling

Non-linear Transform of Features

Convolution

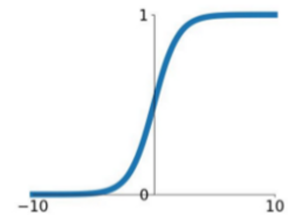


$$(I * K)_{xy} = \sum_{i=1}^{h} \sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y+j-1}$$

Activation function

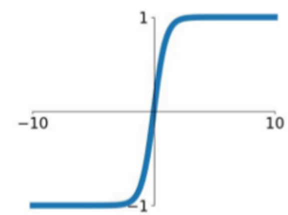
Sigmoid

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



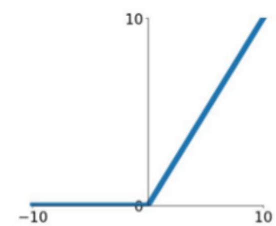
tanh

$\tanh(x)$

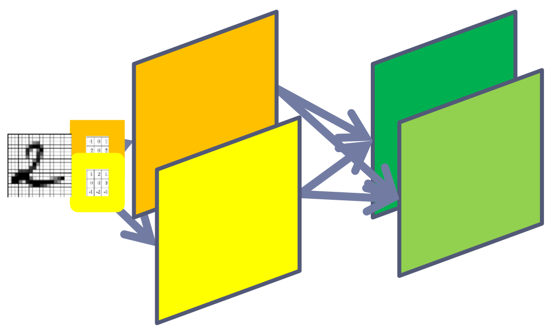


ReLU

$\max(0, x)$

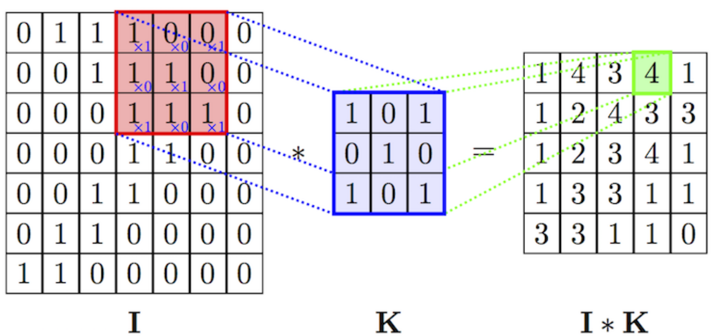


Nonlinearity: HOG vs. CNN



$$Mag = \sqrt{{g_x}^2 + {g_y}^2}$$

$$Ang = arctan\frac{{g_x}^2}{{g_y}^2}$$



Sigmoid

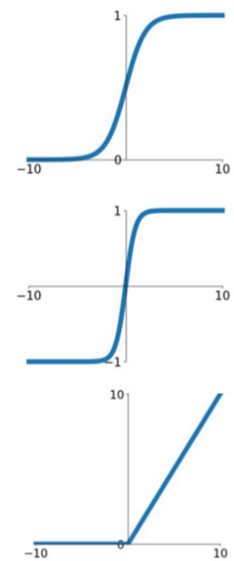
$\sigma(x) = \frac{1}{1+e^{-x}}$

tanh

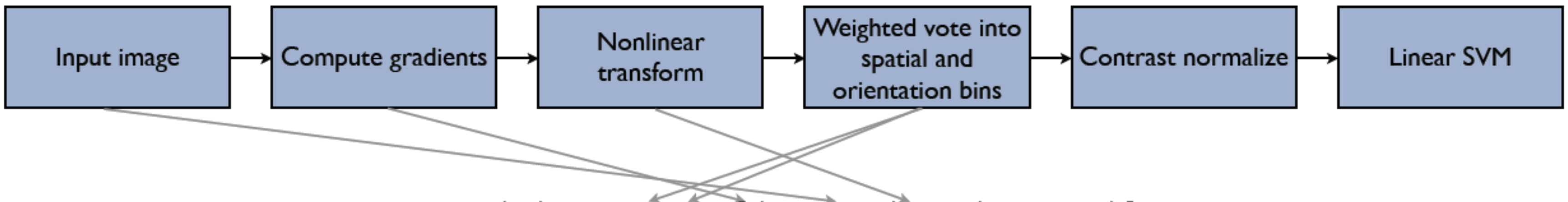
$\tanh(x)$

ReLU

$\max(0, x)$



HOG: Linear Transform of Pixels



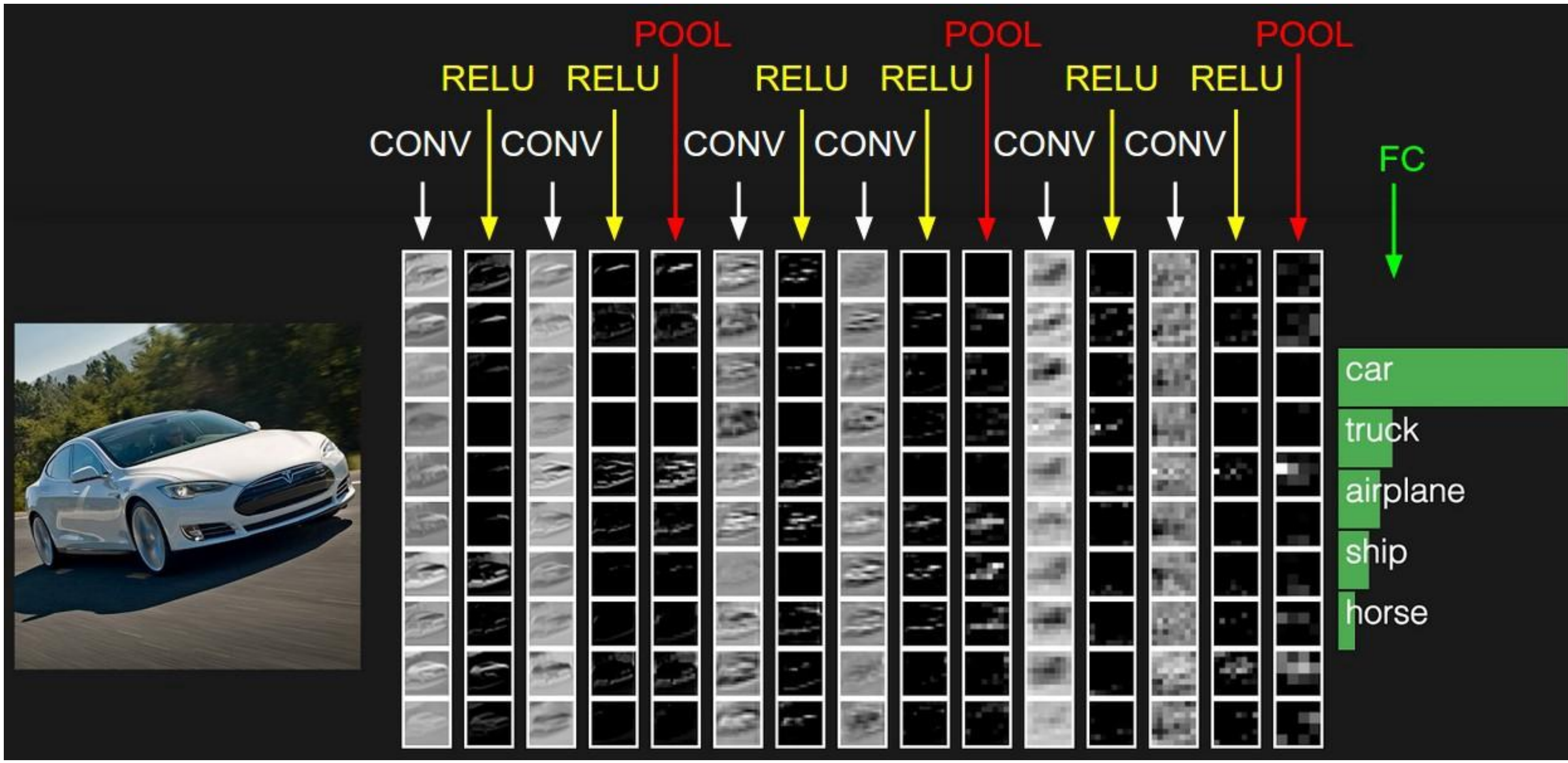
$$\Phi_f(\mathbf{x}) = \mathbf{D}\mathbf{b} * [(\mathbf{g}_f * \mathbf{x}) \odot (\mathbf{g}_f * \mathbf{x})]$$

Figure 1. An illustration of the HOG feature extraction process and how each component maps to our reformulation. Gradient computation is achieved through convolution with a bank of oriented edge filters. The nonlinear transform is the pointwise squaring of the gradient responses which removes sensitivity to edge contrast and increases edge bandwidth. Histogramming can be expressed as blurring with a box filter followed by downsampling.

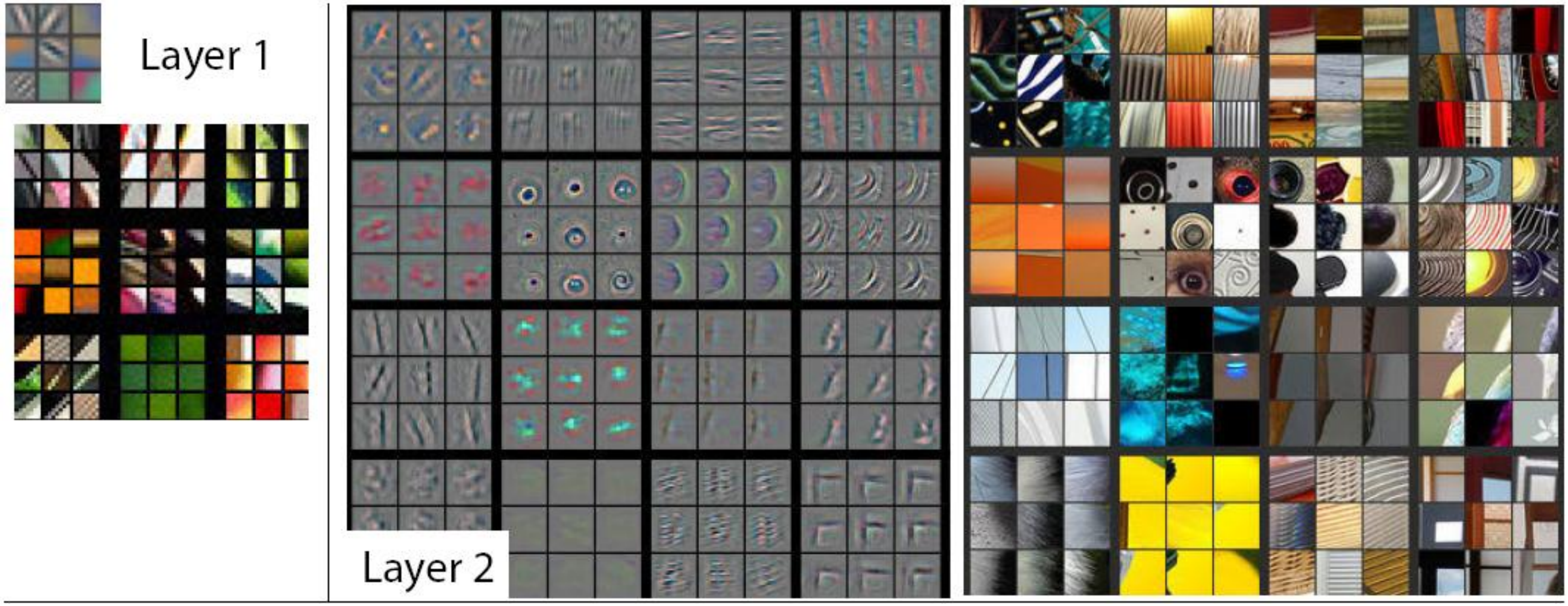
x – Input image
$\mathbf{g}_f$ – Oriented edge filter
b – Blur operator
D – Sparse selection matrix for pooling/histogram

(Hilton Bristow and Simon Lucey,
Why do linear SVMs trained on HOG features perform so well?, 2014)

Nonlinearity

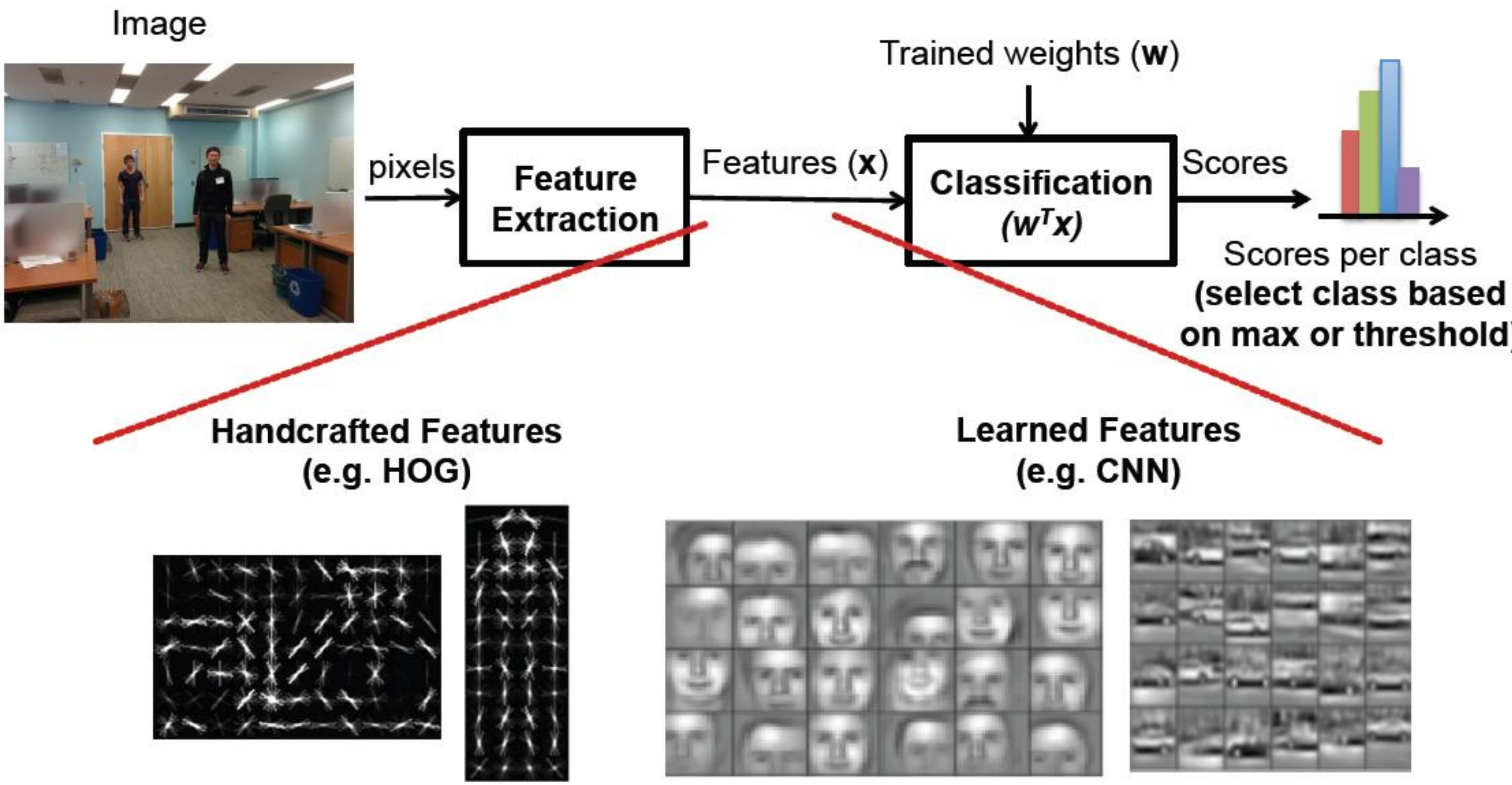


Why Deep?



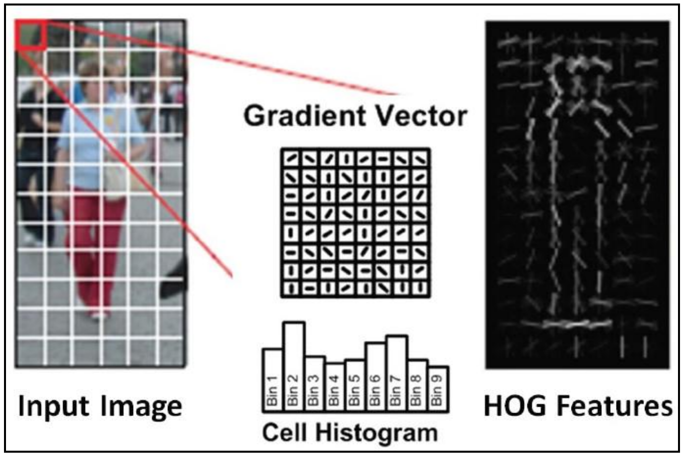
Matthew D. Zeiler and Rob Fergus, Visualizing and Understanding Convolutional Networks, 2014

PR: Feat Engineering vs. Feat. Learning



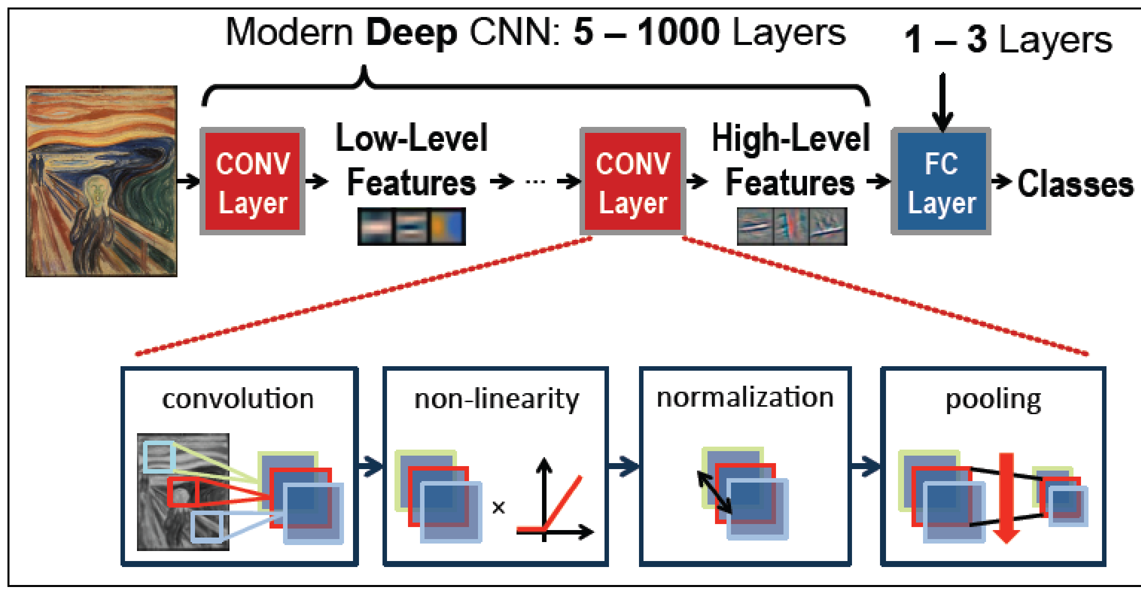
A. Suleiman, Y. H. Chen, J. Emer and V. Sze, "Towards closing the energy gap between HOG and CNN features for embedded vision," 2017.

"Deep" Feature Learning vs. "Shallow" Feature Engineering



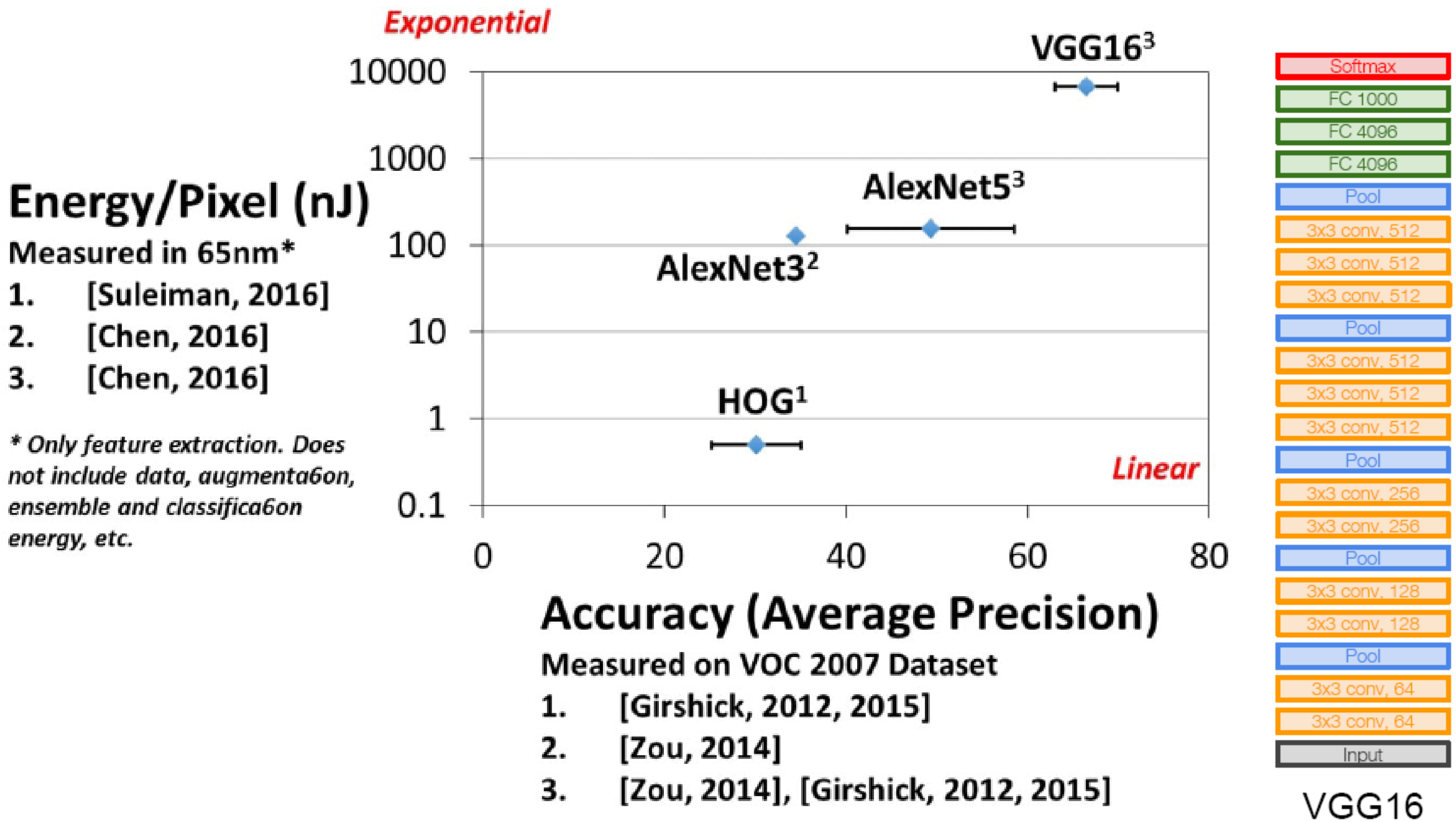
"Shallow"

"Deep"



A. Suleiman, Y. H. Chen, J. Emer and V. Sze, "Towards closing the energy gap between HOG and CNN features for embedded vision," 2017.

Performance:
Feat. Learning vs. Feat. Engineering



A. Suleiman, Y. H. Chen, J. Emer and V. Sze, "Towards closing the energy gap between HOG and CNN features for embedded vision," 2017.

“Hand-Crafted” Feature Extraction

Domain Specific Feature

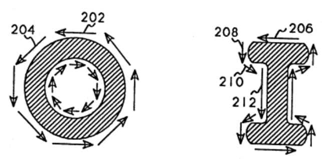


FIG. 2

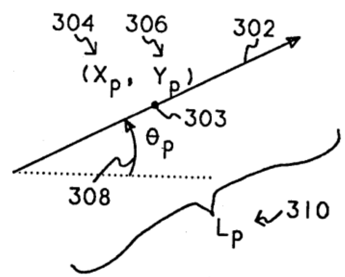
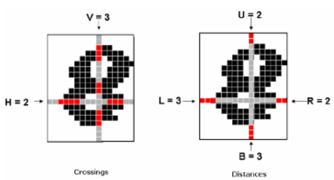
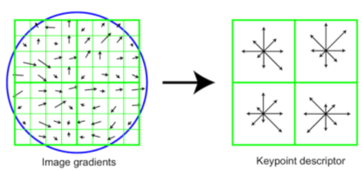
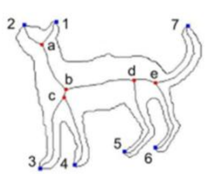
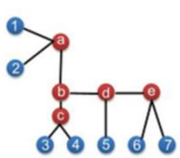


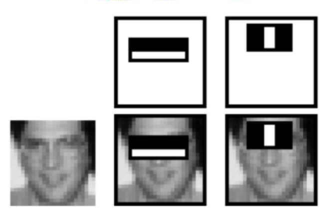
FIG. 3



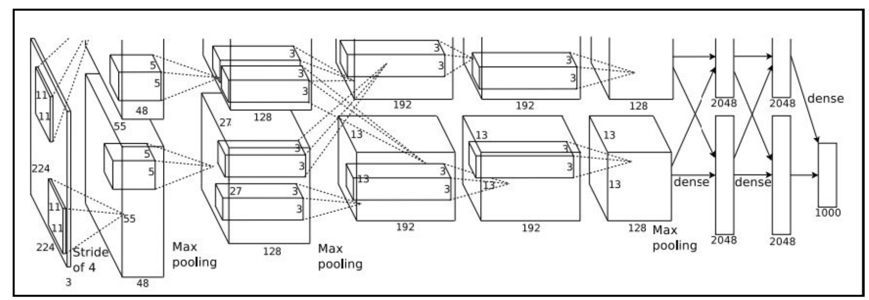


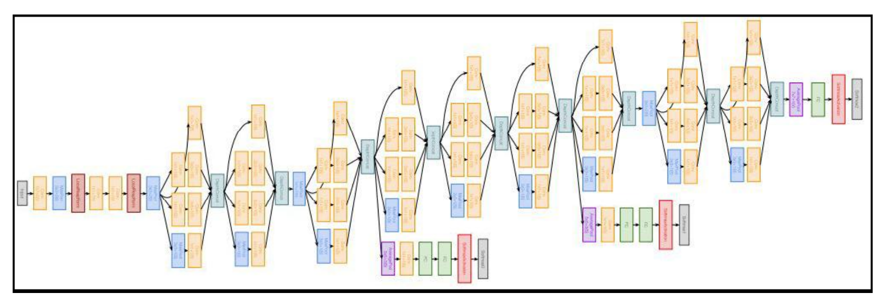






Designed Architecture



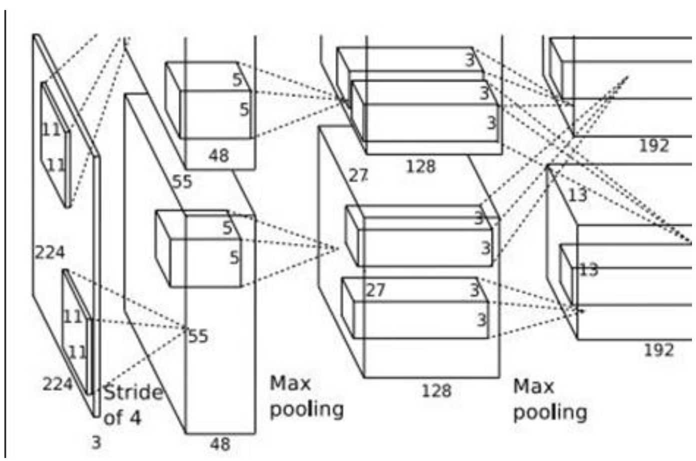


Architecture Design: Speed

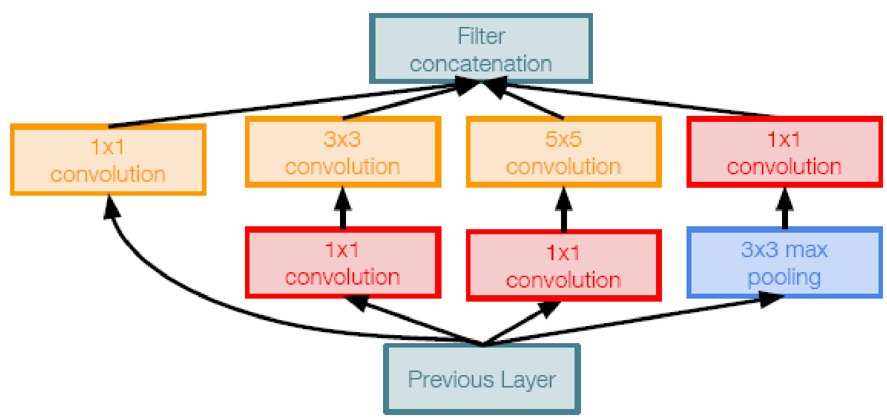
- Simplification

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

- Parallelization

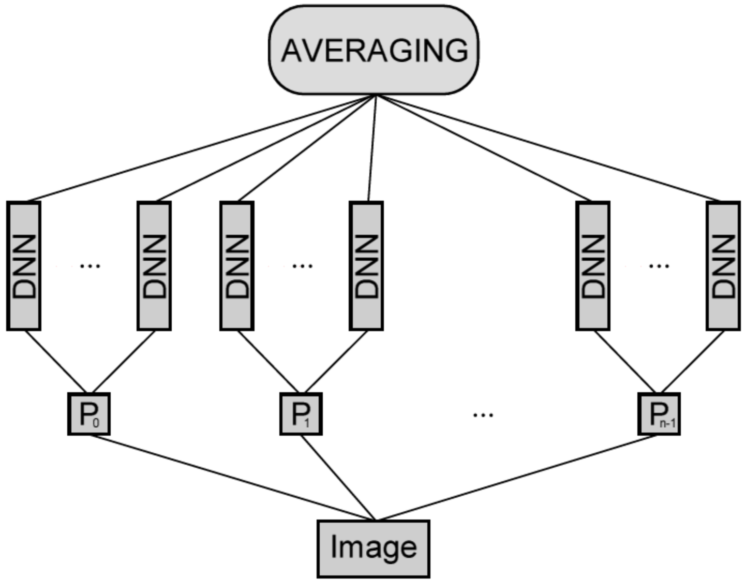


- Hand-design sub-network



Architecture Design: Accuracy

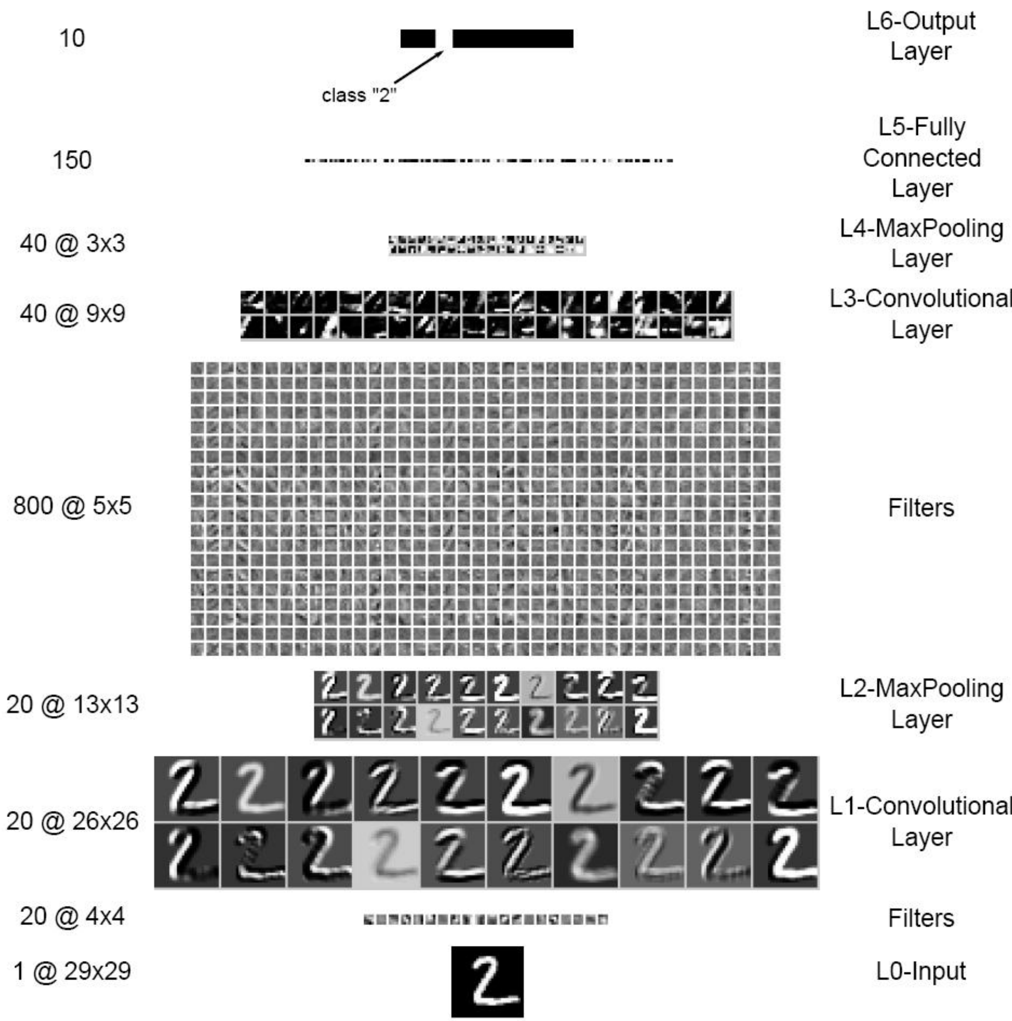
- Multicolumn CNN for MNIST
  - 10, 12, 14, 16, 18, 20 sizes normalization
  - 5 DNN columns per normalization, total of 35 columns
  - 1x29x29-20C4-MP2-40C5-MP3-150N-10N DNNs are trained



- Performance

k-NN	2-layer NN	SVM RAW	LeNet-5	Mul.Col. DNN	SVM HOG
5.0	4.7	1.4	0.95	**0.23**	0.61

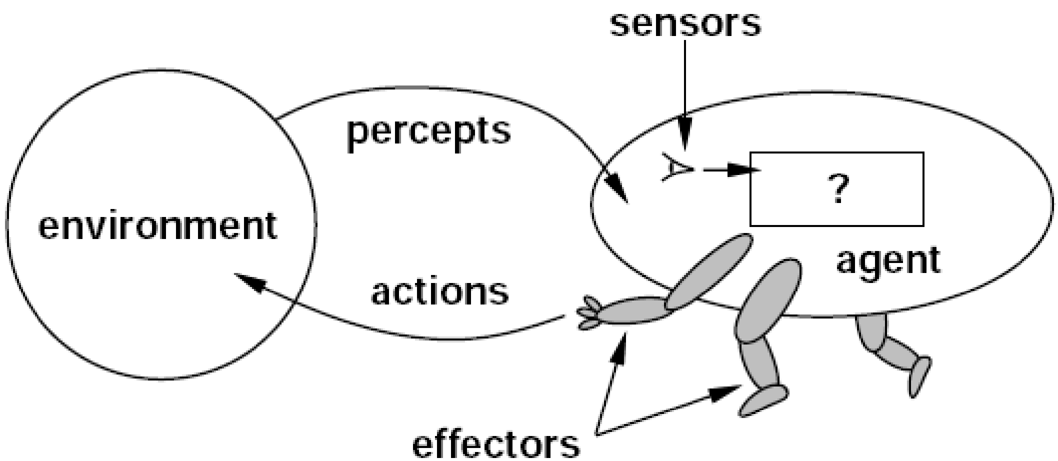
Multi-column Deep CNN for MNIST



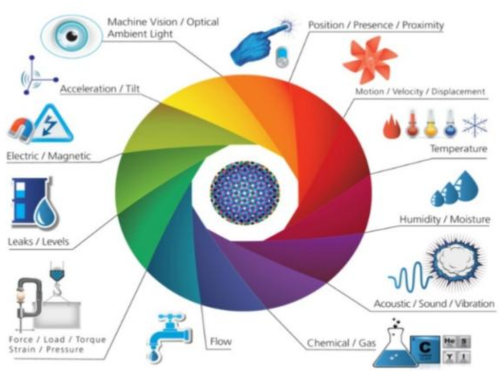
Dan Cireşan, Ueli Meier, Juergen Schmidhuber, Multi-column Deep Neural Networks for Image Classification, 2012

What Next?

AI

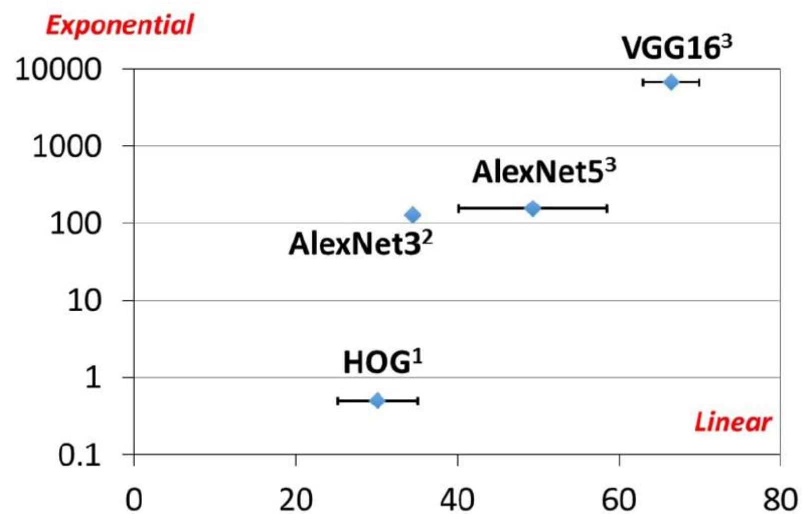


Trillion Sensor World



How to

Feature
Learning
AND/OR
Engineering



Why Pattern Recognition is Hard

- **<u>Text detection</u>**

- **<u>Character recognition</u>**

  PLAYA CERRADA

  RECENTE ATAQUE DE TIBURON

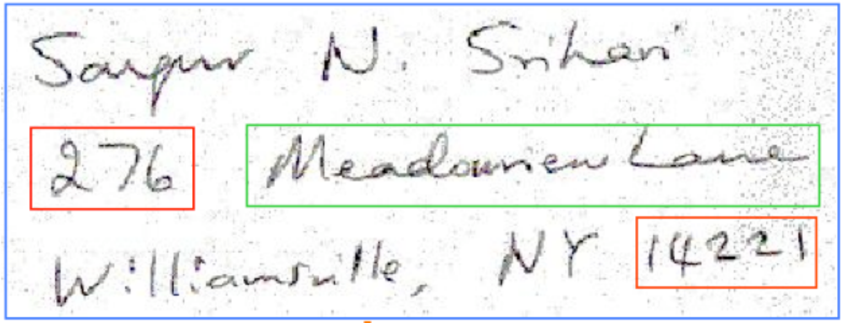
- **<u>Language translation</u>**

  BEACH CLOSED

  RECENT ATTACK OF SHARK

Why Pattern Recognition is Hard

Street address



Database query

ZIP Code: 14221
Primary number: 276

Records Retrieved

Lexicon entry (Street name)	ZIP+4 add-on
AMHERSTON DR	7006
BELVOIR RD	
CADMAN DR	
CLEARFIELD DR	
FORESTVIEW DR	
HARDING RD	7111
HUNTERS LN	3330
MCNAIR RD	3718
MEADOWVIEW LN	3557
OLD LYME DR	2250
RANCH TRL	2340
RANCH TRL W	2246
SHERBROOKE AVE	3421
SUNDOWN TRL	2242
TENNYSON TER	5916

Recognizer choice (after lex. expansion)

Address encoding

ZIP+4: 142213557

Why Pattern Recognition is Hard

Ground Truth – Word Recognition

Dataset Images

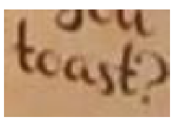
word_1.png

word_2.png

word_3.png

word_4.png

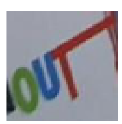
word_5.png



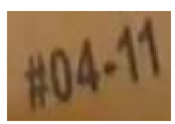
word_6.png

word_7.png

word_8.png

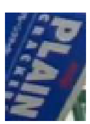


Word_9.png

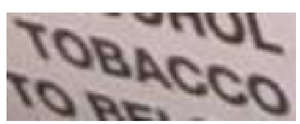


word_10.png

word_11.png



word_12.png



word_13.png

Ground Truth transcription

```
word_1.png, "$500"
word_2.png, "who"
word_3.png, "SMRT"
word_4.png, "COACH"
word_5.png, "FALL"
word_6.png, "toast?"
word_7.png, "SEASON!"
word_8.png, "HUMP"
word_9.png, "OUT"
word_10.png, "#04-11"
word_11.png, "NEW"
word_12.png, "PLAIN"
word_13.png, "TOBACCO"
...
```

gt.txt

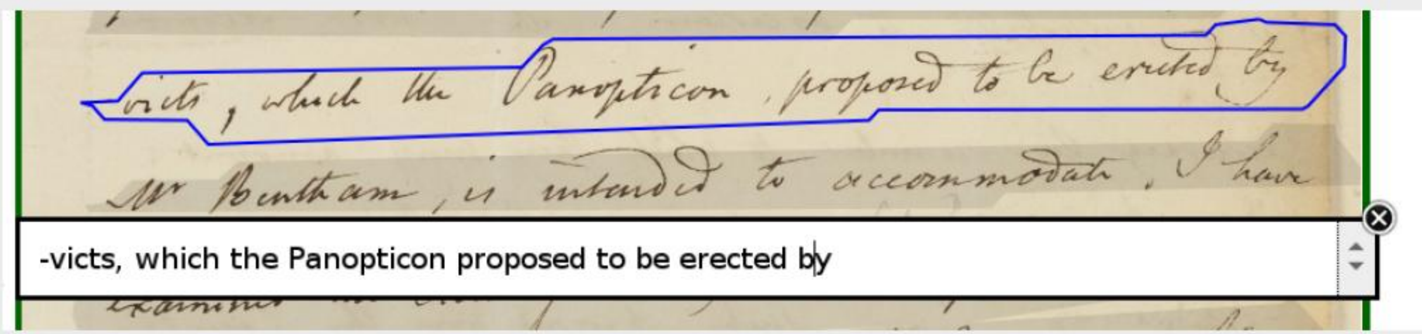
Ground Truth location (ONLY Challenge 4)

```
word_1.png,0,18,88,0,90,50,2,68
word_2.png,23,13,229,0,207,138,0,152
word_3.png,8,22,152,0,146,57,0,90
word_4.png,0,96,153,0,178,40,26,136
word_5.png,0,50,116,0,152,83,3,122
word_6.png,1,0,63,16,62,41,0,26
word_7.png,0,5,82,0,83,24,1,29
word_8.png,9,8,349,0,340,83,0,91
word_9.png,0,41,86,0,101,56,16,97
word_19.png,0,21,70,0,76,29,6,50
word_11.png,0,4,91,0,91,28,0,32
word_12.png,0,90,41,0,72,6,27,96
word_13.png,0,0,100,24,105,39,4,15
...
```

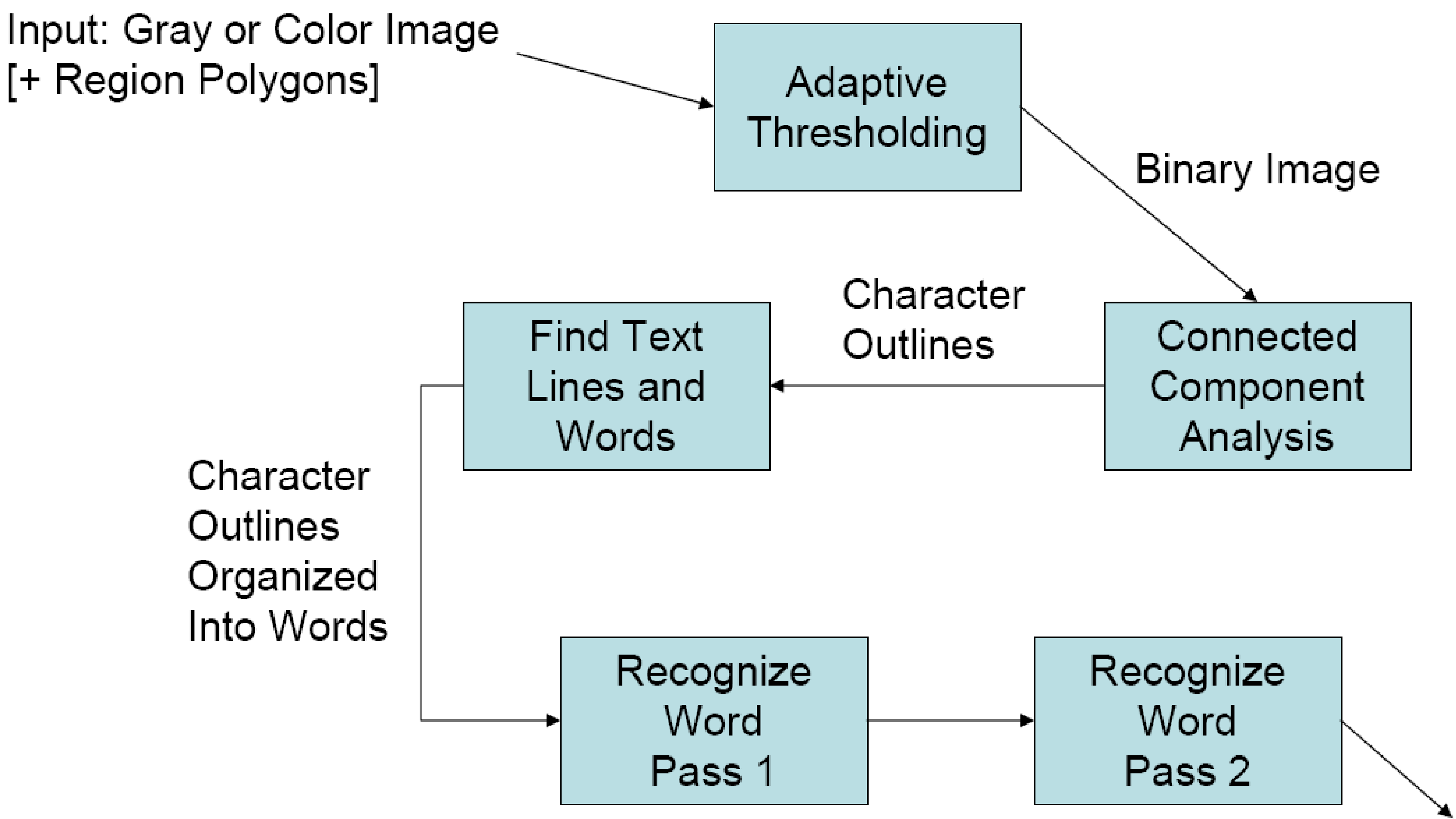
coords.txt

Why Pattern Recognition is Hard

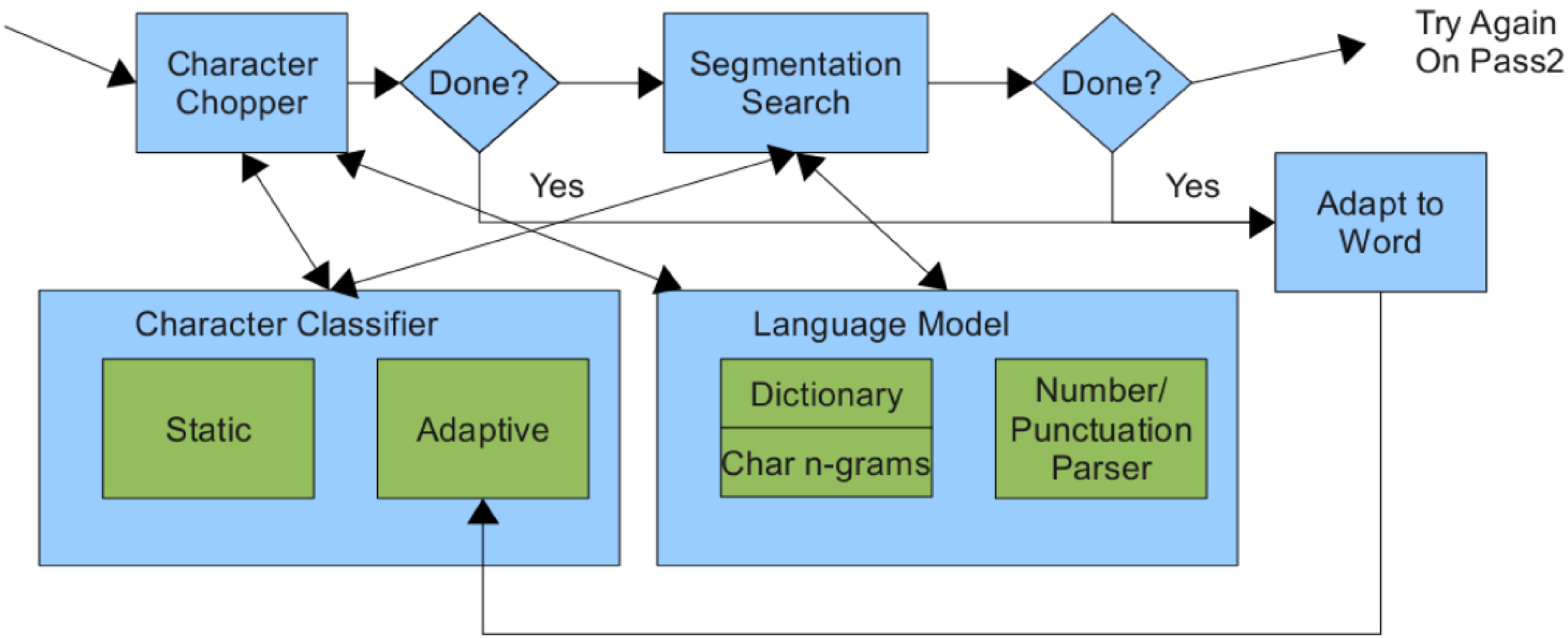
- **Interactive Handwritten Text Recognition:** the user and the system interact for obtaining the correct transcript.



Tesseract OCR

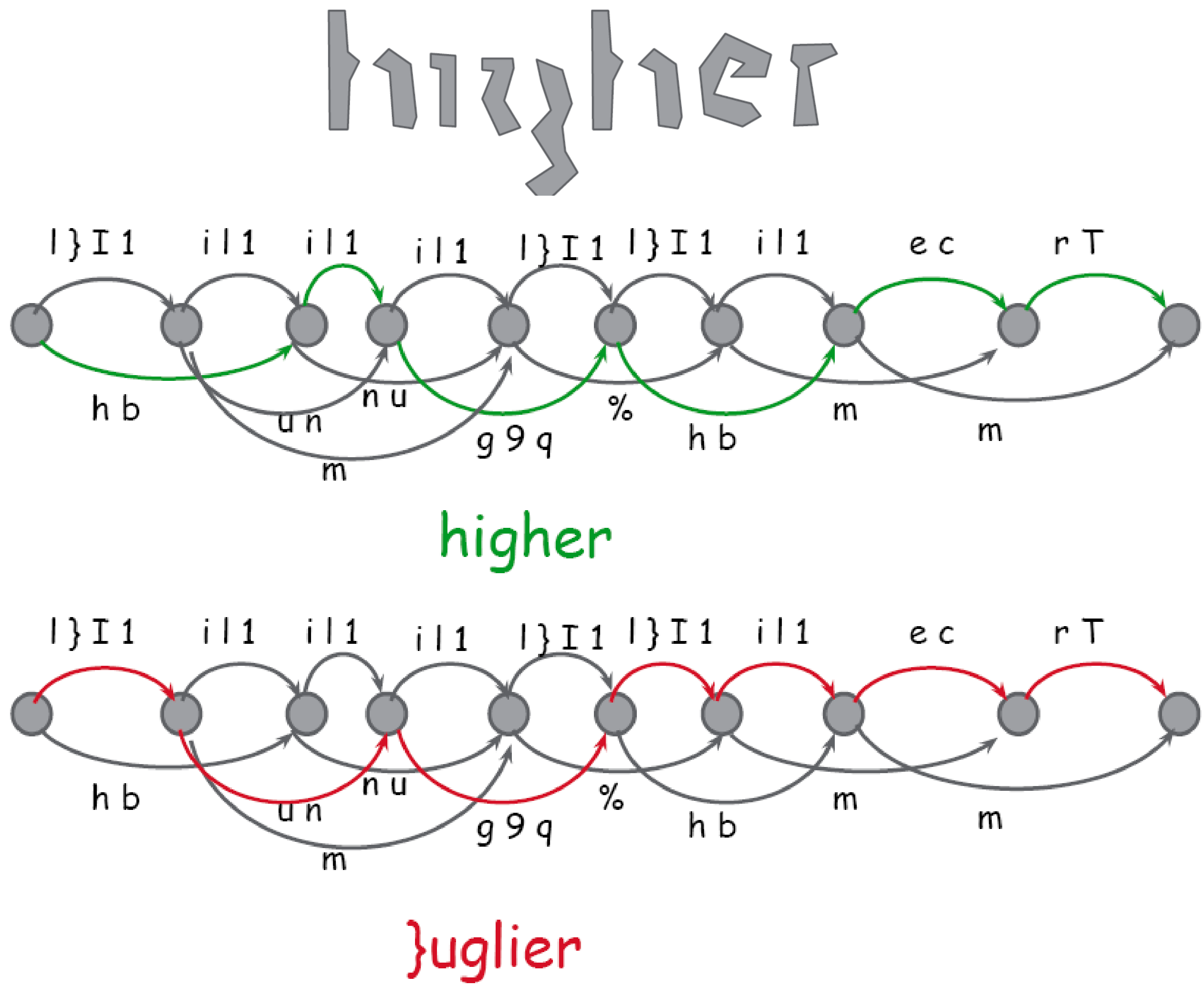


Tesseract Word Recognition

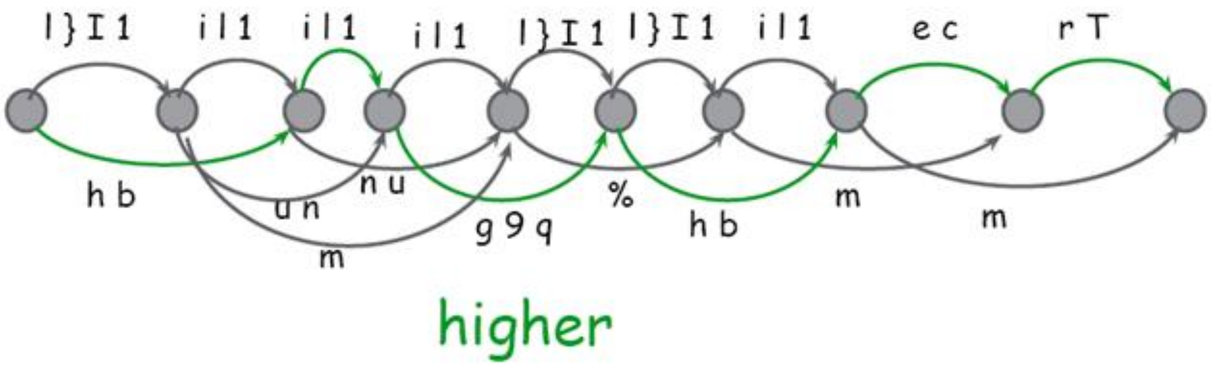


Character Over-segmentation

Segmentation Graph



OCR With (Lexical) Context



$$\hat{C} = \underset{C}{\operatorname{argmax}}\, \mathrm{p}(signal|C) \cdot \mathrm{p}(C)$$

$$p(signal|C) = \prod_i p(s_i|c_i) = \prod_i \frac{\exp(output(s_i|c_i))}{\sum_j \exp(output(s_j|c_j))}$$

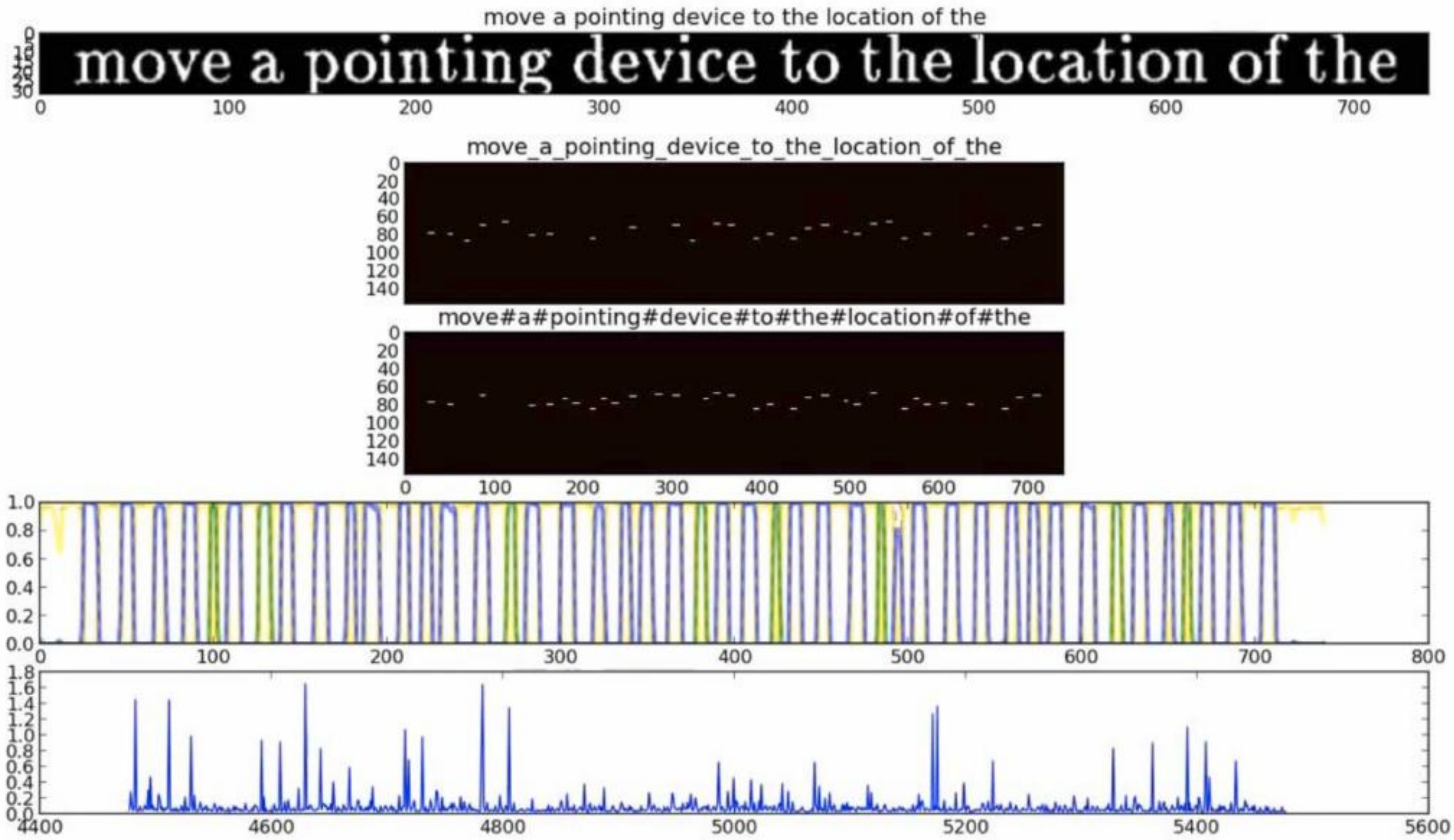
$$p(C) = \prod_i p(c_i|\phi(h(i))) = \prod_i p(c_i|c_1 c_2 \ldots c_{i-1})$$

length penalty

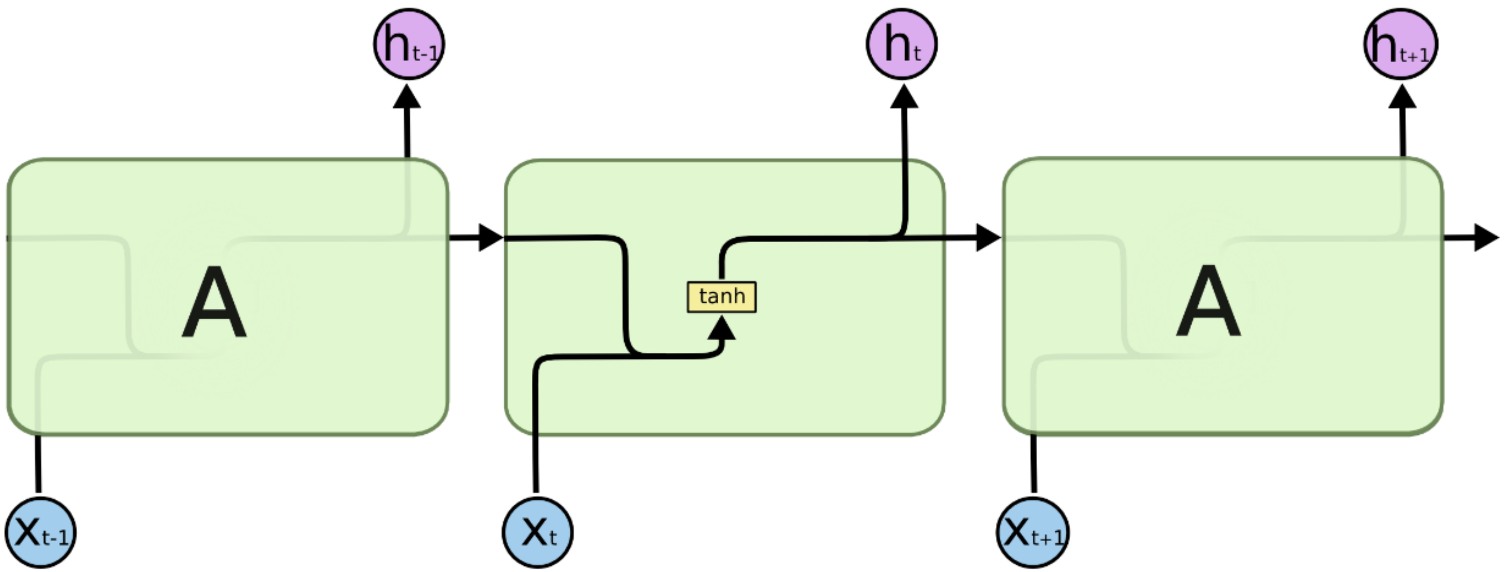
context (LM) weight

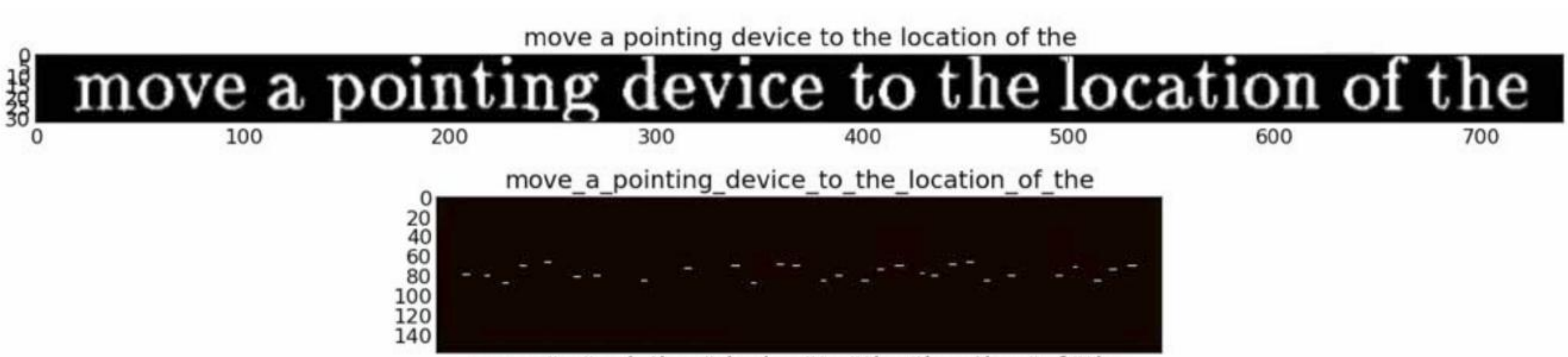
$$\hat{C} = \arg max \sum_i (log(p(s_i|c_i)) + \gamma \cdot log(p(c_i|\phi(h_n(i)))) + \delta)$$

OCR With Long-Short Term Memory

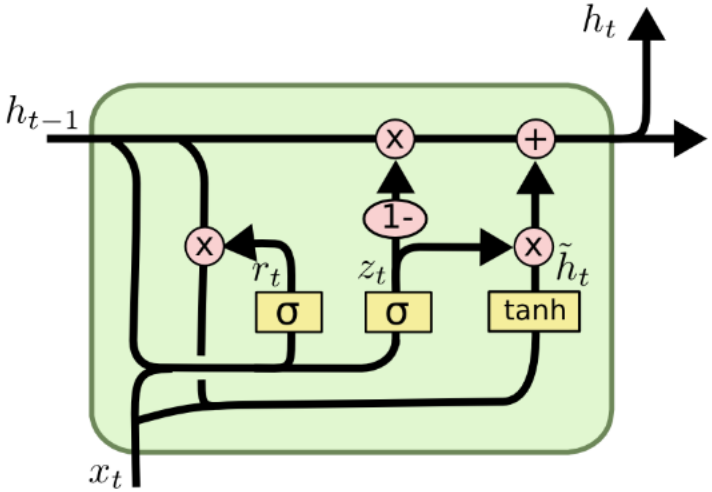


Recurrent Neuron Networks





Long-Short Term Memory

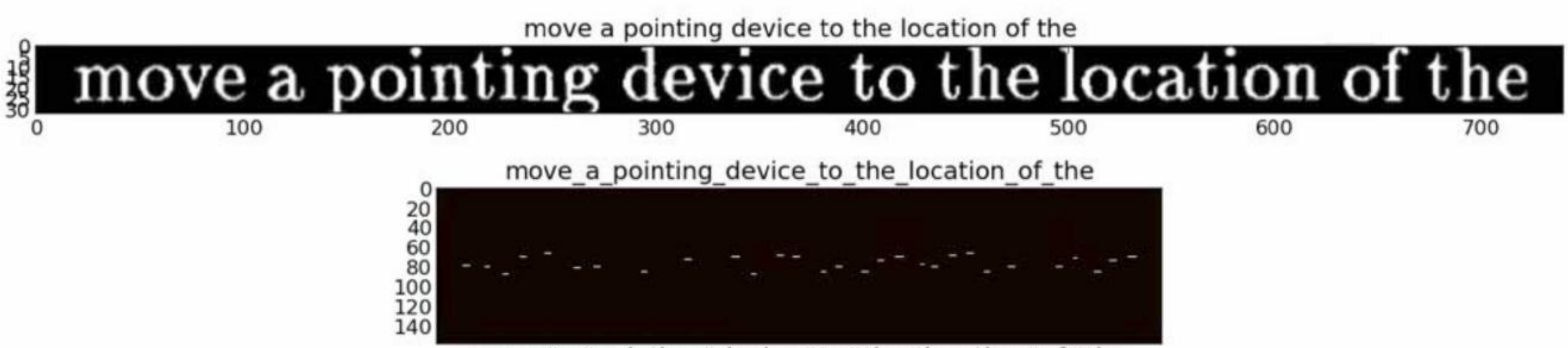


$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

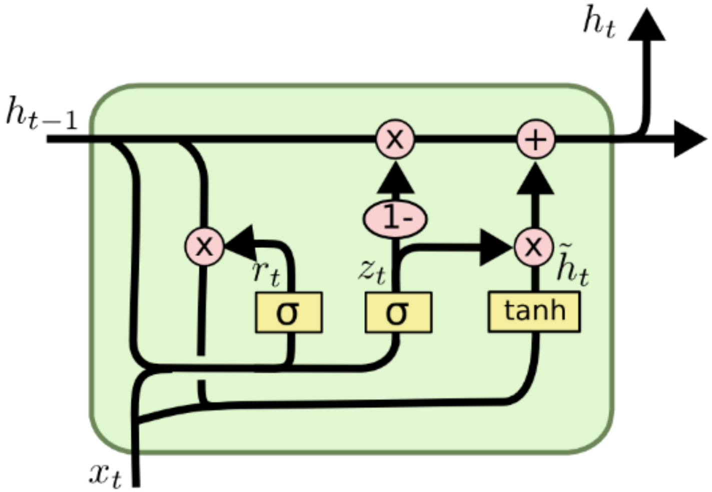
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$\boxed{h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t}$$



LSTM vs. Language Model?

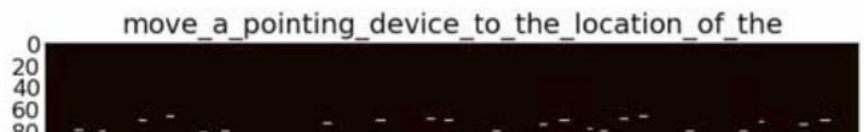


$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$\boxed{h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t}$$



$$\hat{C} = \arg max \sum_i \boxed{(log(p(s_i|c_i)) + \gamma \cdot log(p(c_i|\phi(h_n(i)))) + \delta)}$$

"There is no one model that works best for every problem"

Reference

- Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11):2278-2324, November 1998.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in NIPS, 2012.
- A. Suleiman, Y. H. Chen, J. Emer and V. Sze, "Towards closing the energy gap between HOG and CNN features for embedded vision," *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore, MD, 2017, pp. 1-4.
- D. Ciregan, U. Meier and J. Schmidhuber, "Multi-column deep neural networks for image classification," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, 2012, pp. 3642-3649.
- https://www.learnopencv.com/histogram-of-oriented-gradients/
- https://wiki.tum.de/display/lfdv/Convolutional+Neural+Networks
- A Beginner's Guide To Understanding Convolutional Neural Networks
- http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- …