Tutorial Session

Introduction to ML

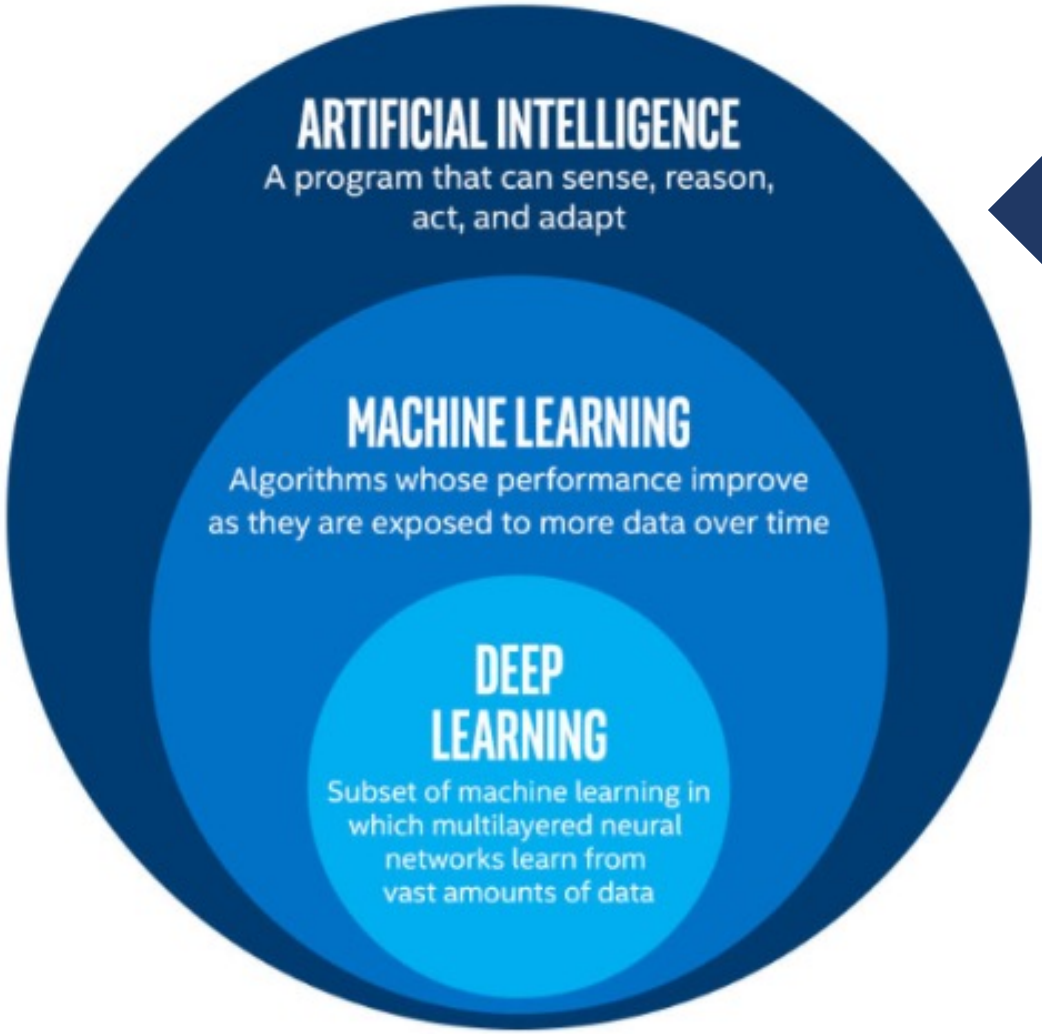
Mu-Chieh (Laura) Ko
NOAA/AOML/HRD and UM/CIMAS
UM/ECE

Why do you want to do ML?

What ML technique to use?

How to prepare the data?

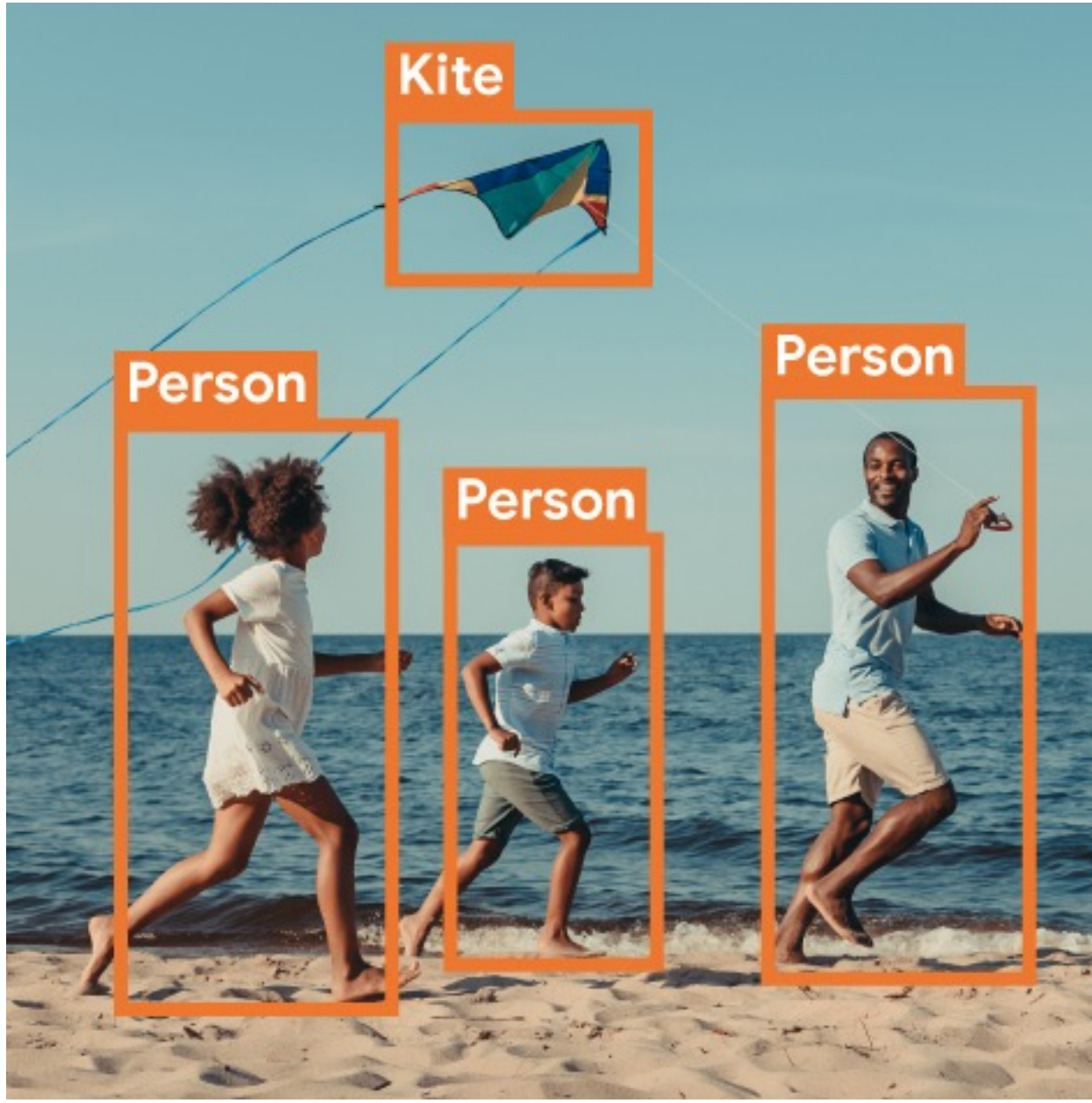
What is ML



- Robotic System
- Computer Vision
- Audio Processing

Why AI/ML?

How do you program an object?



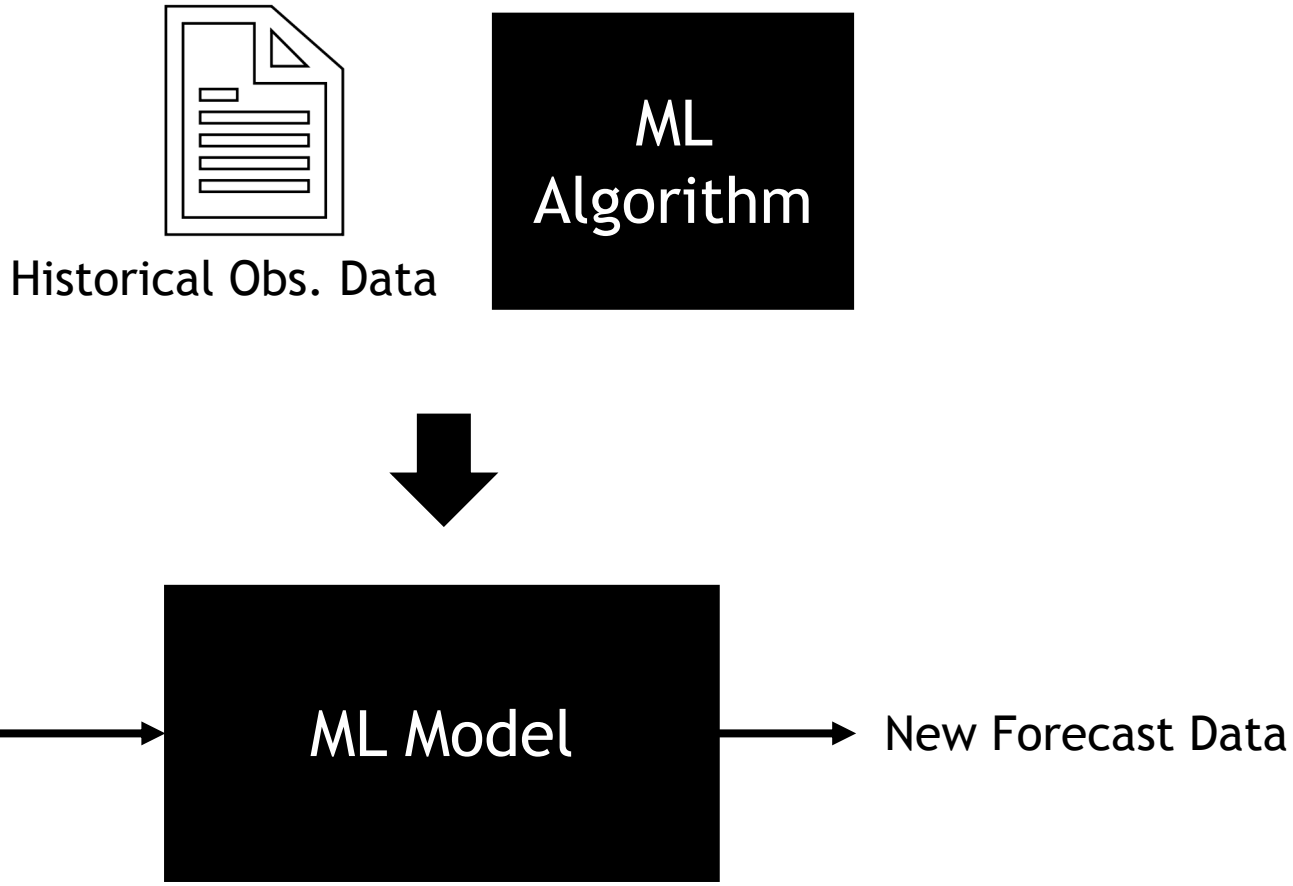
Credit: https://blog.tensorflow.org/2020/07/tensorflow-2-meets-object-detection-api.html

Why AI/ML

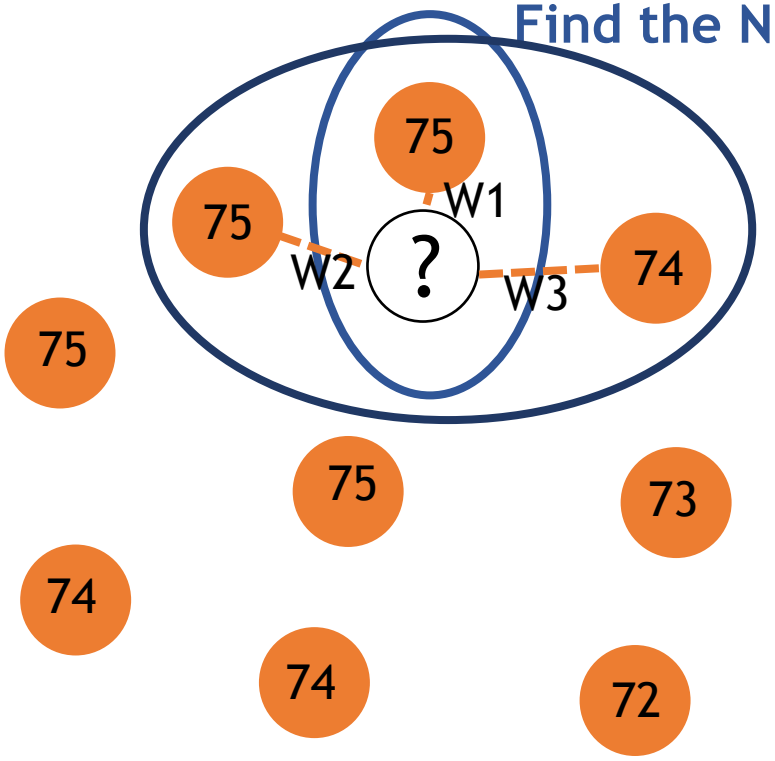
Programming based on physics, dynamics, etc.

New Observation Data → Numerical Model → New Forecast Data

Why AI/ML


Historical Obs. Data
ML Algorithm
ML Model
New Observation Data
New Forecast Data

k-Nearest Neighbors



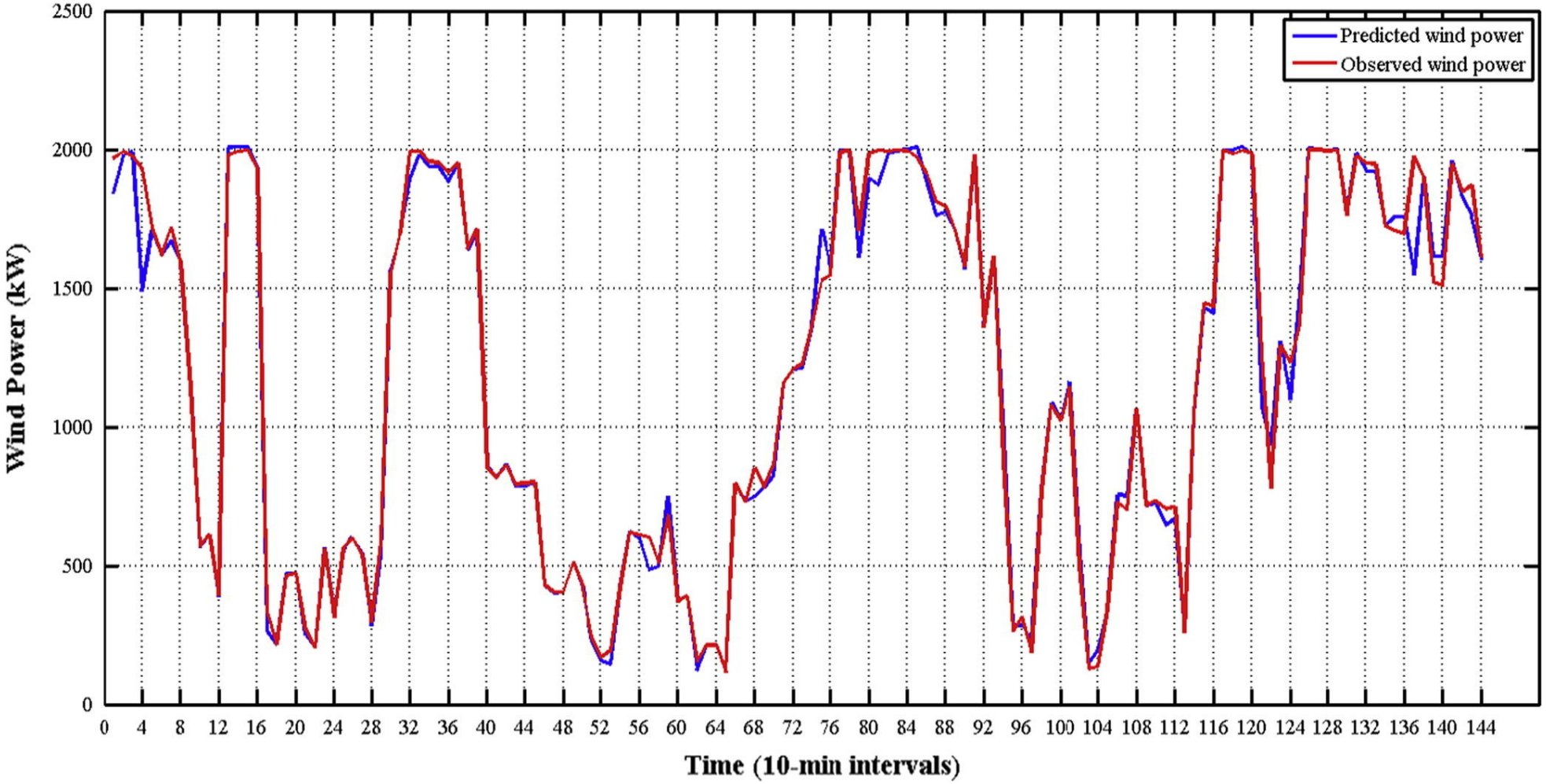
Find the Nearest Neighbor (?) =75

Find K Nearest Neighbors (?) =mean(75, 75, 74)= 74.7

Weighted

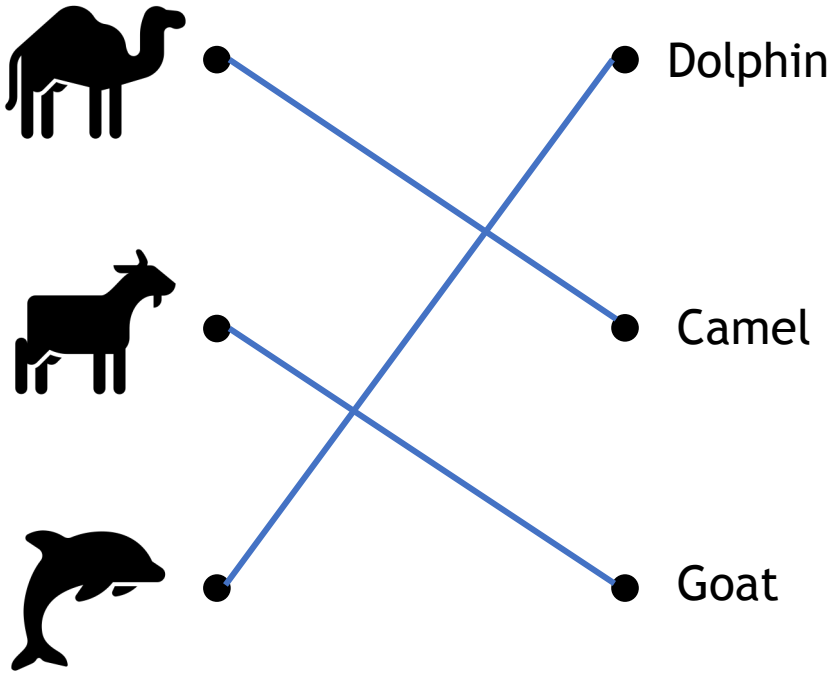
(?) =W1*75+ W2*75+ W3*74
=74.9

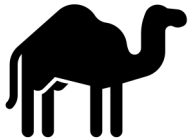
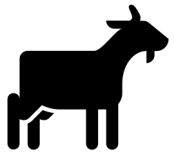
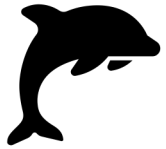
k-Nearest Neighbors



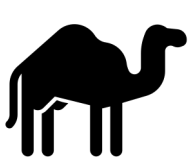
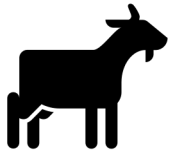
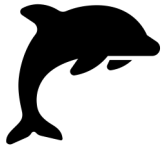
Yesibudak et al. https://doi.org/10.1016/j.enconman.2016.12.094

Decision Trees

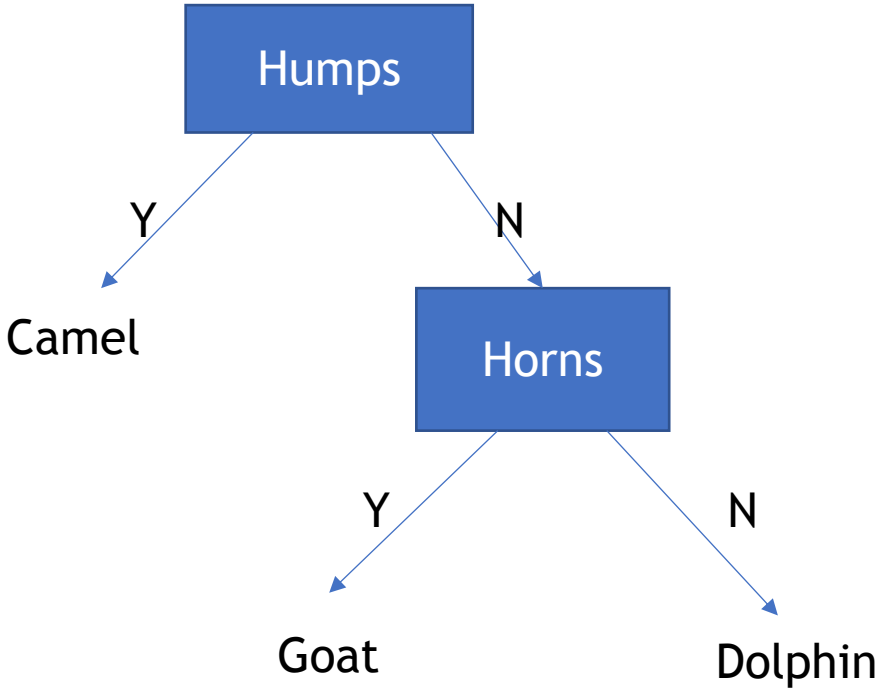


			
Humps?	yes	no	no
Horns?	no	yes	no
Fins?	no	no	yes

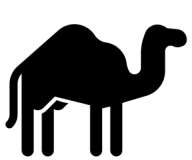
Decision Trees

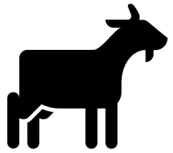




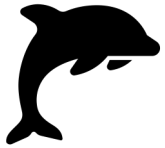
Humps?	yes	no	no
Horns?	no	yes	no
Fins?	no	no	yes



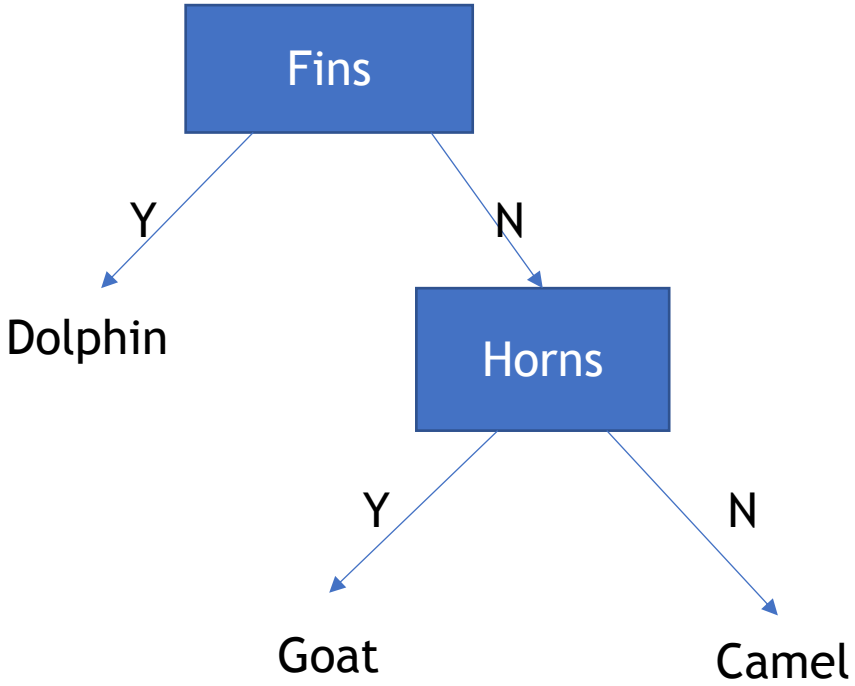
Decision Trees



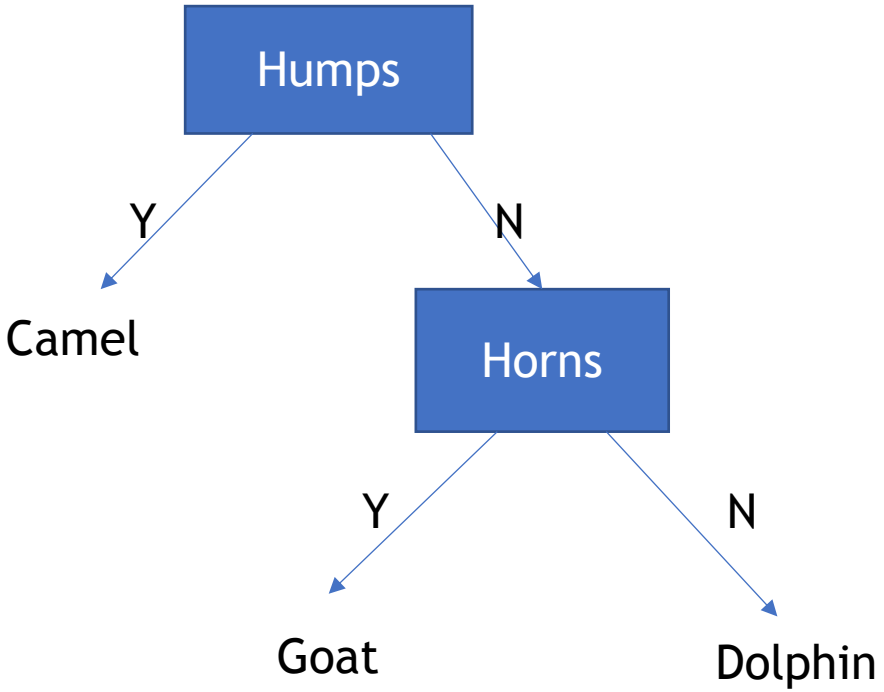


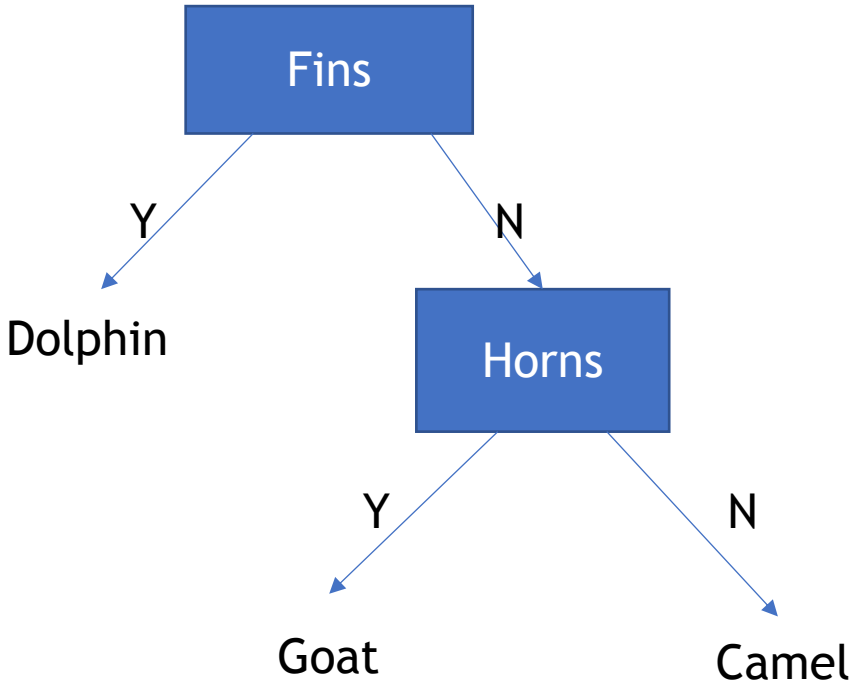


Humps?	yes	no	no
Horns?	no	yes	no
Fins?	no	no	yes



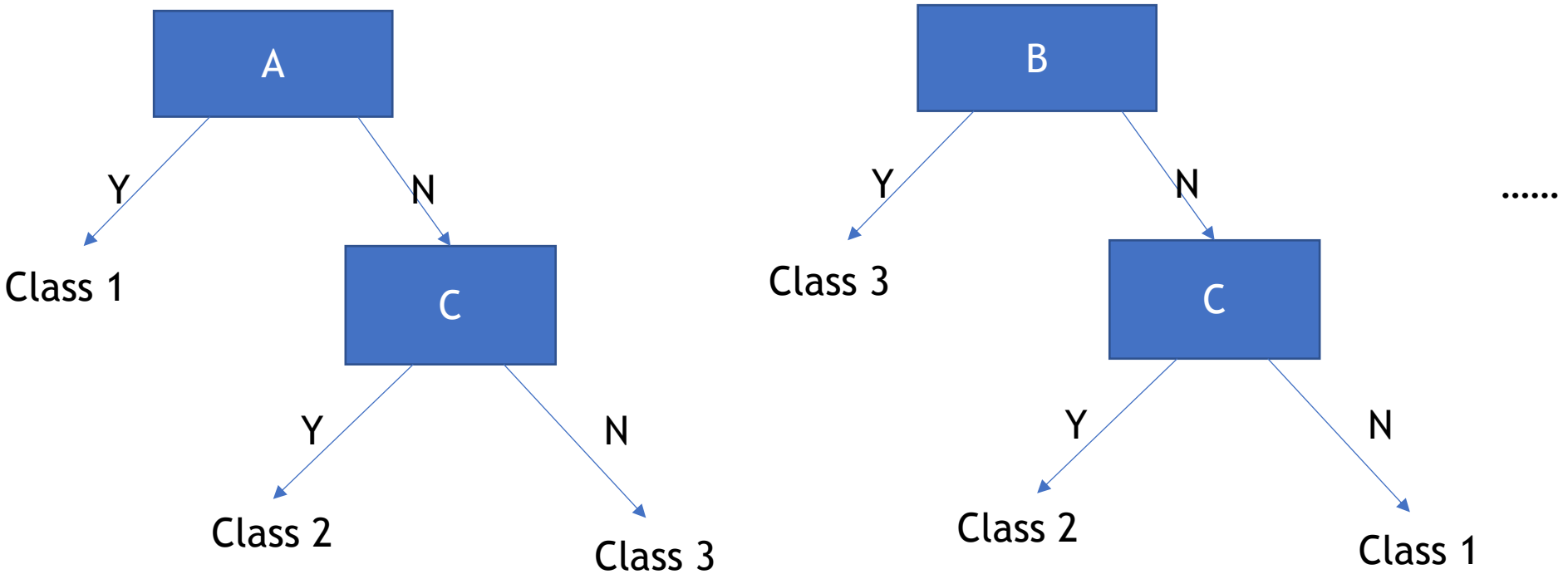
Decision Trees



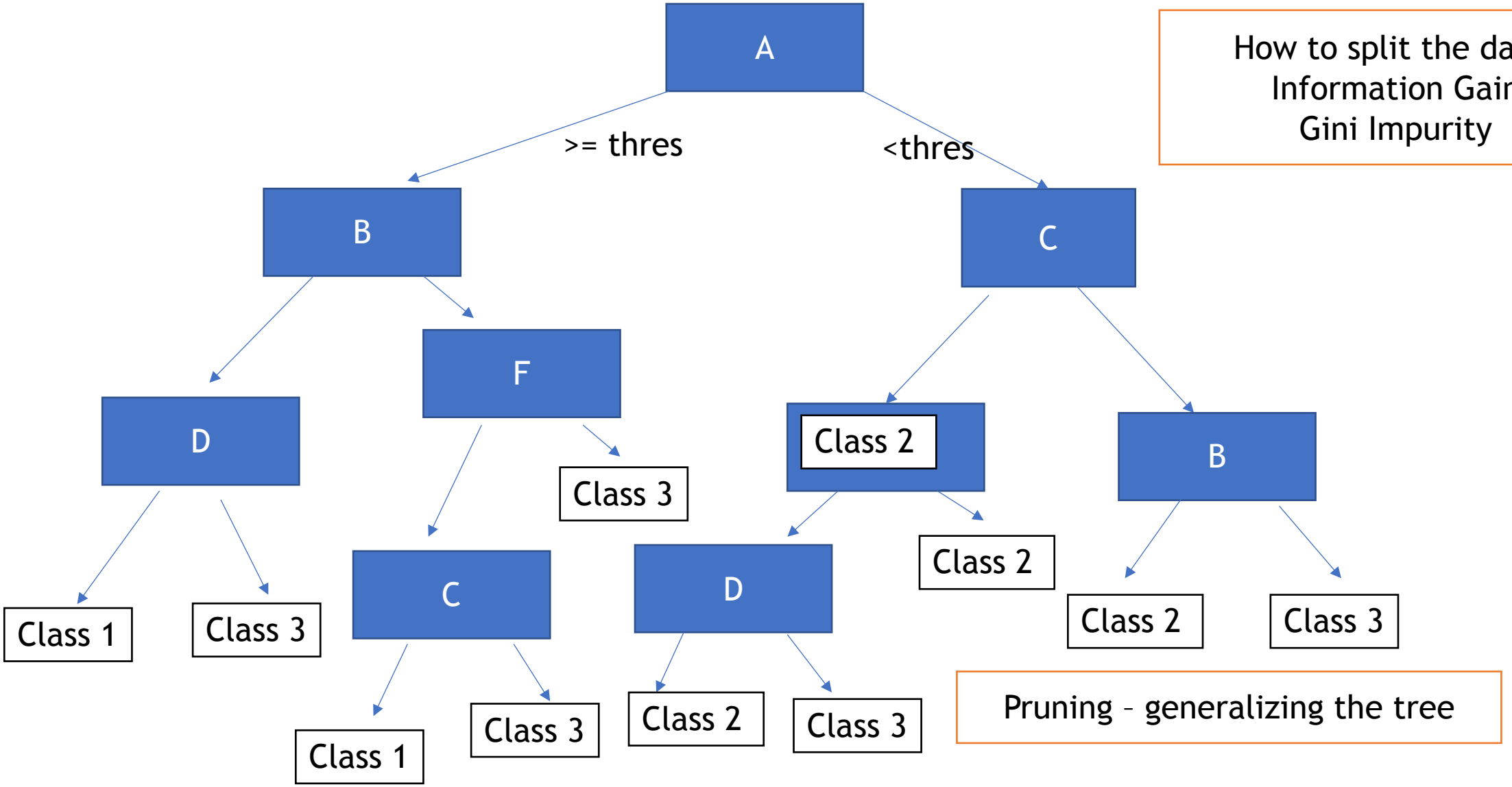


Decision Trees

Idea: Randomizing training data or selected features



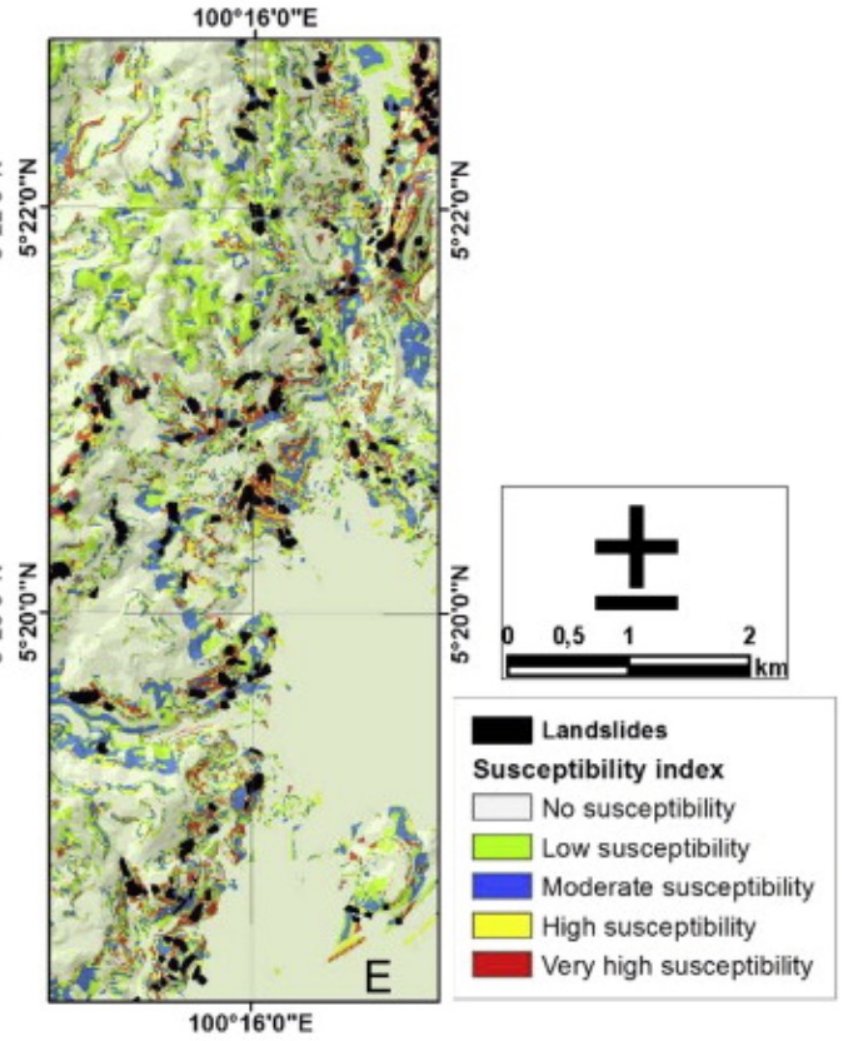
Decision Trees



How to split the data?
Information Gain
Gini Impurity

Pruning – generalizing the tree

Decision Trees



ɔn Index

g

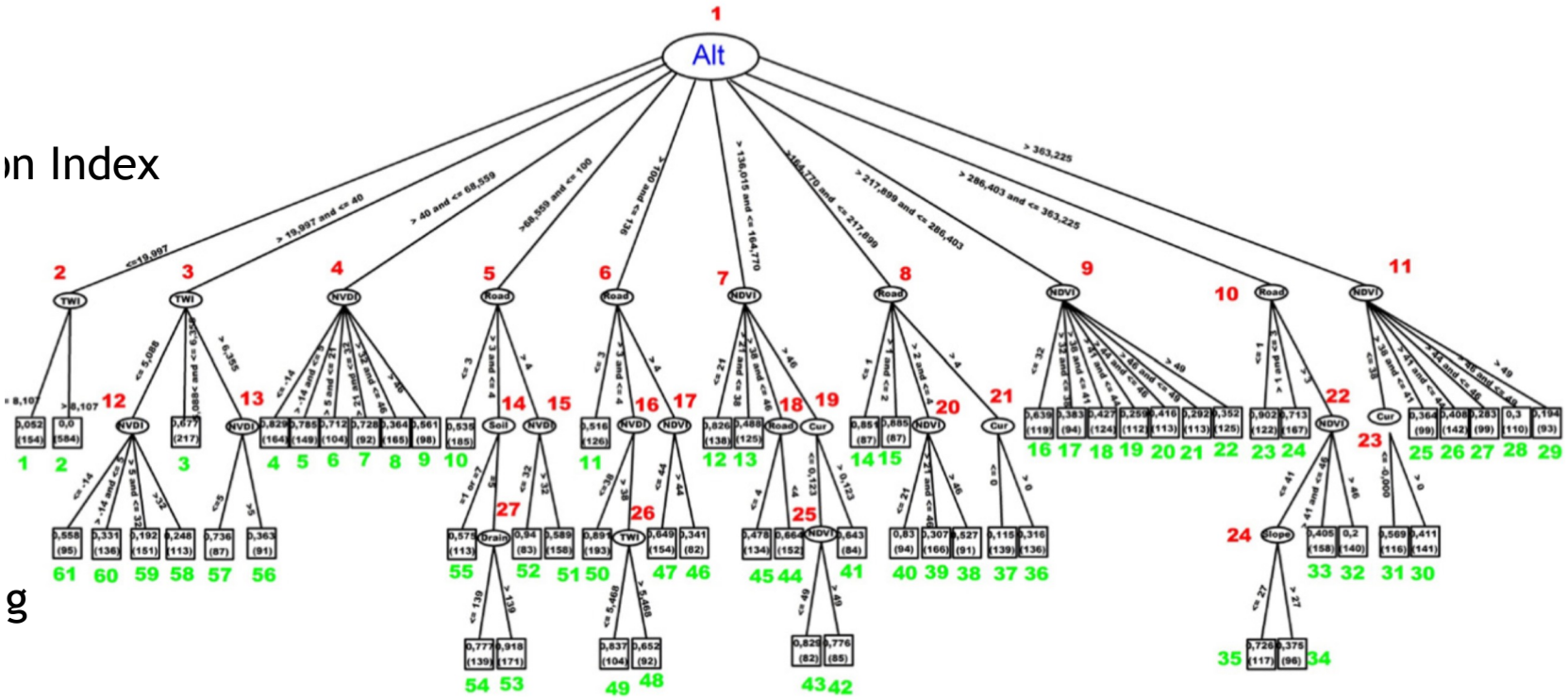
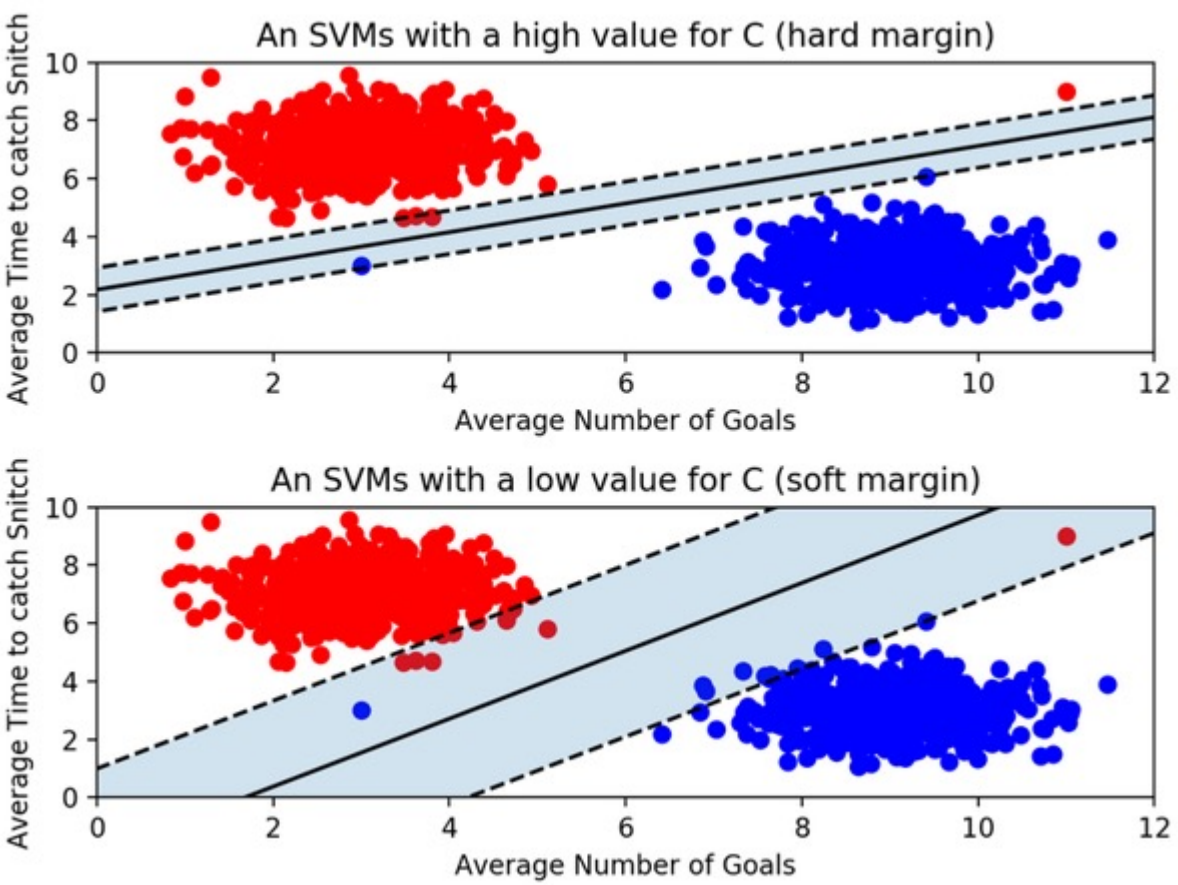
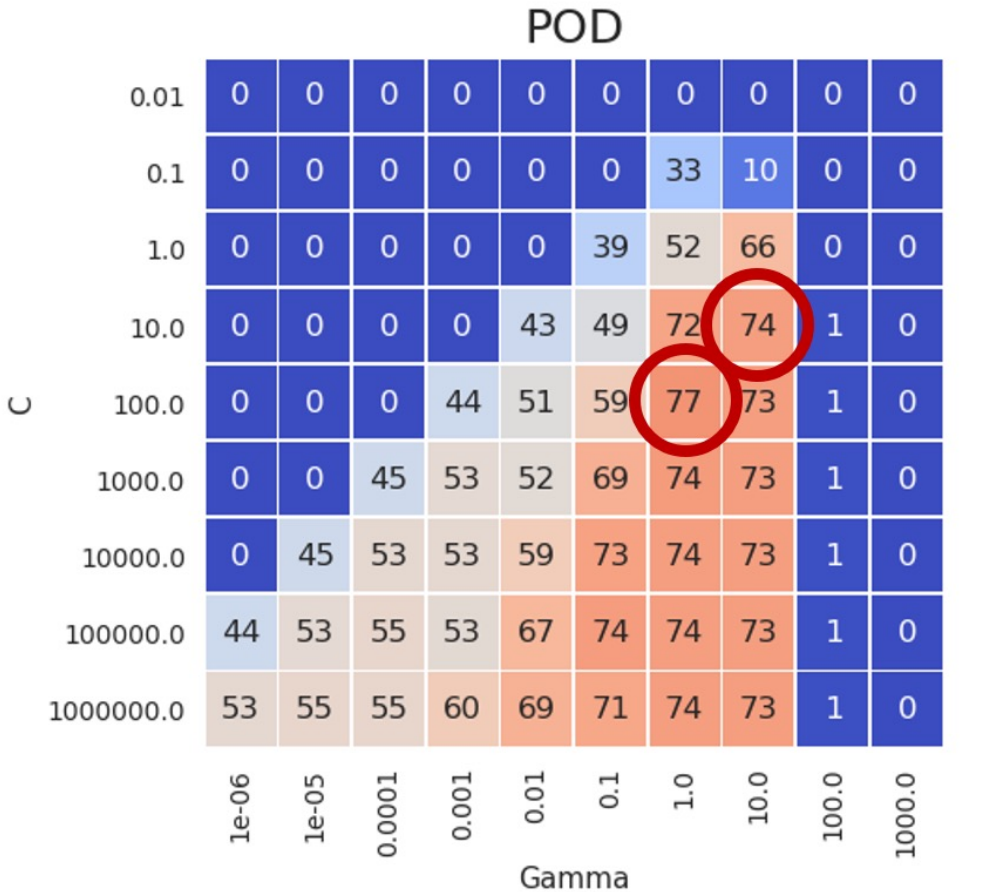


Fig. 4. An example view of decision tree for model 5.

Pradhan https://doi.org/10.1016/j.cageo.2012.08.023

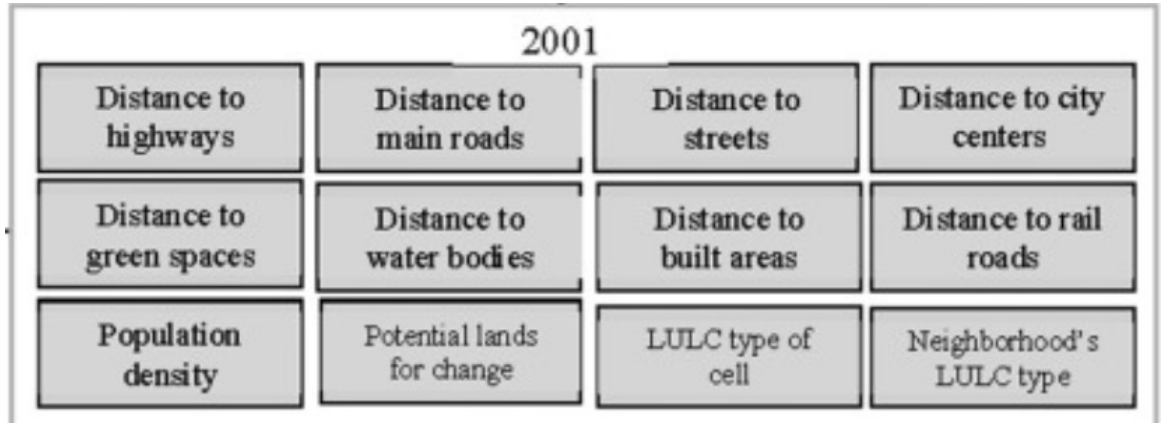
Support Vector Machine

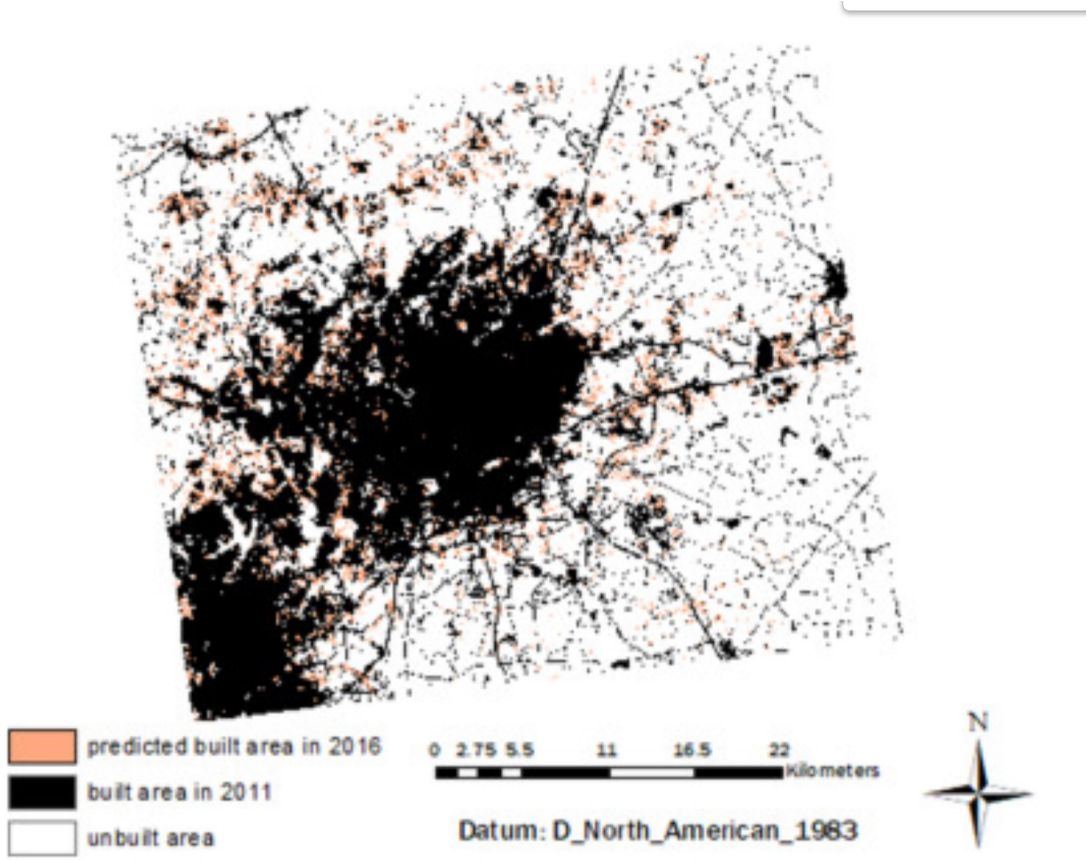




Ko 2020, https://ams.confex.com/ams/2020Annual/meetingapp.cgi/Paper/364456

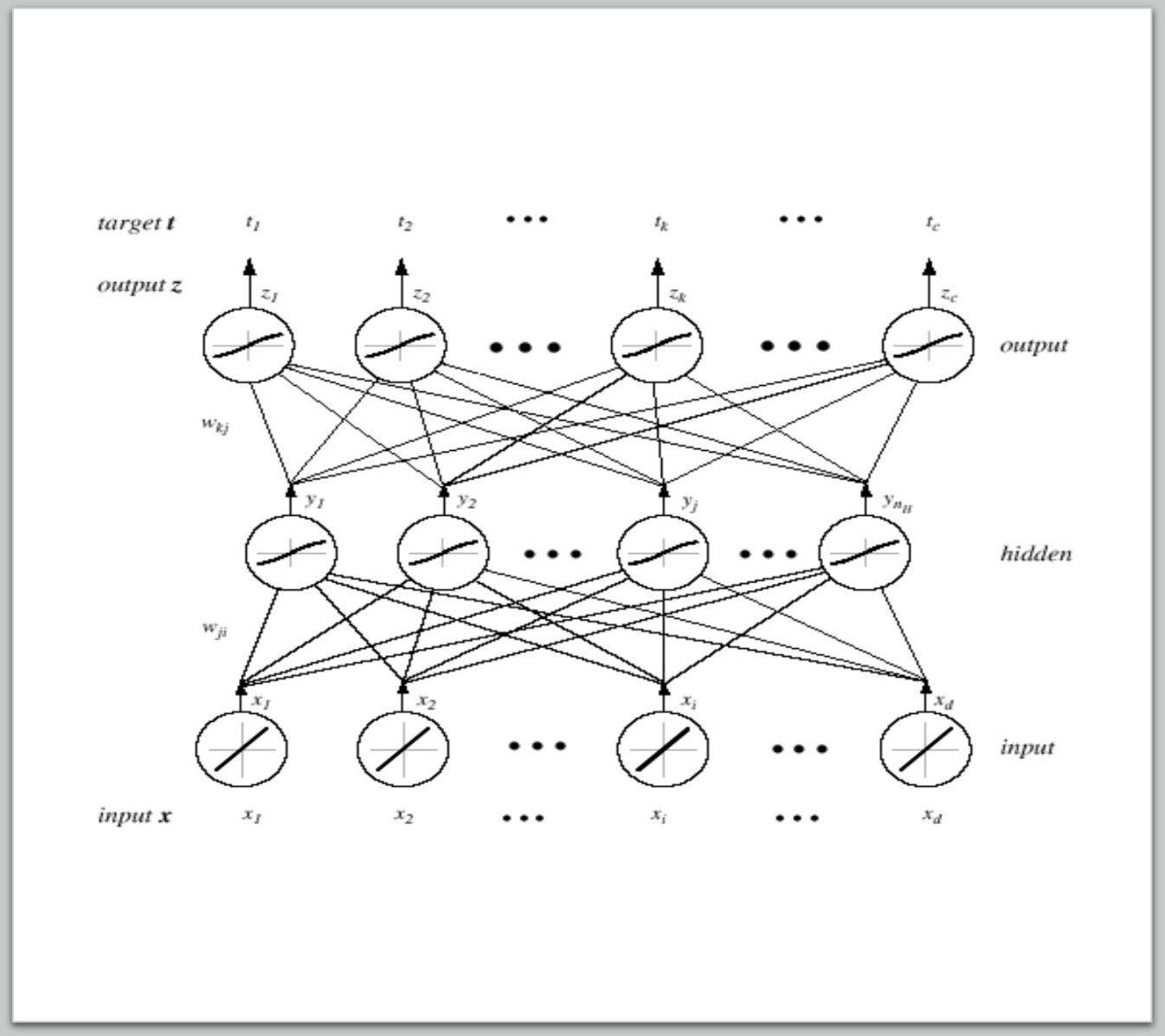
Support Vector Machine



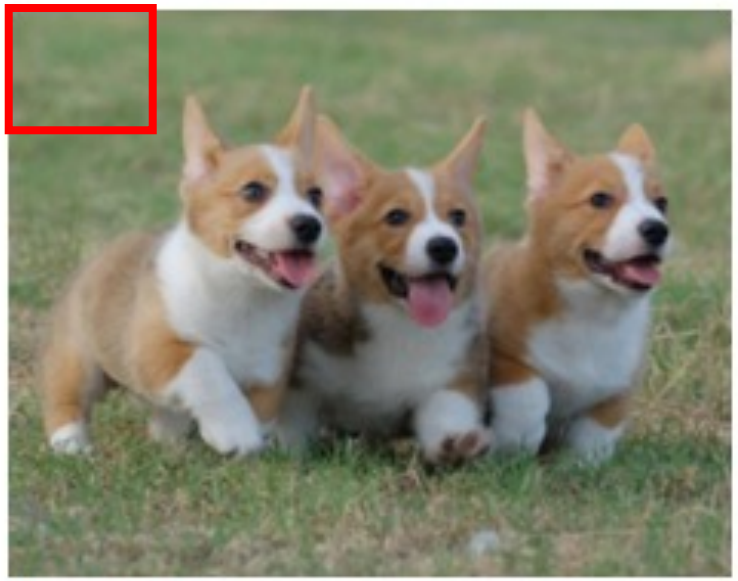


Karimi et al. (2019) https://doi.org/10.1016/j.compenvurbsys.2019.01.001

Artificial Neural Networks



Convolutional Neural Networks (CNN)



What We See

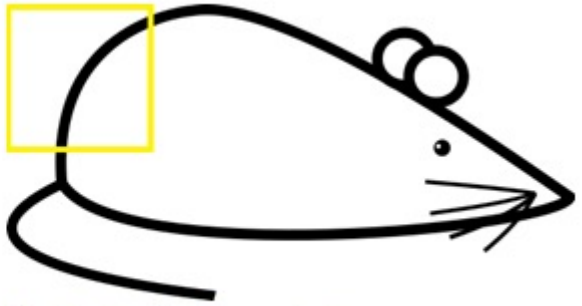
```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48
```

What Computers See

Reference: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

CNN

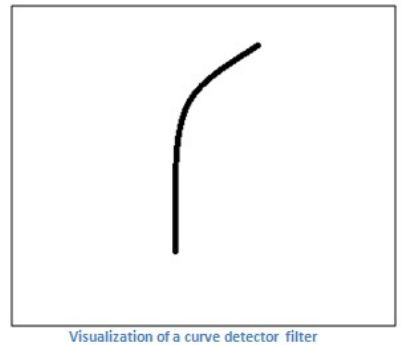
Original image



Visualization of the filter on the image

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter



Visualization of a curve detector filter

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

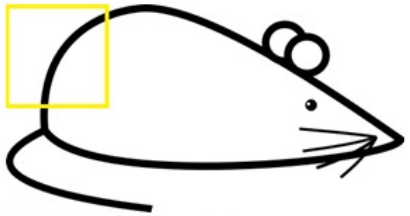
*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

CNN



Visualization of the filter on the image

receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

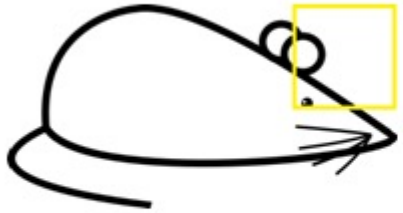
Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)



Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

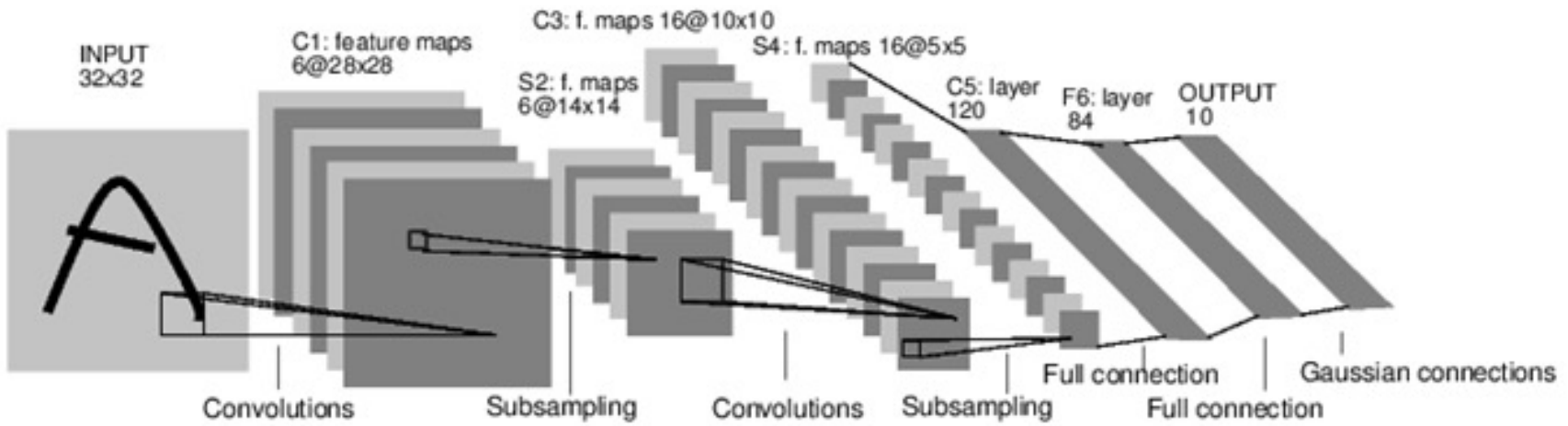
*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = 0

CNN



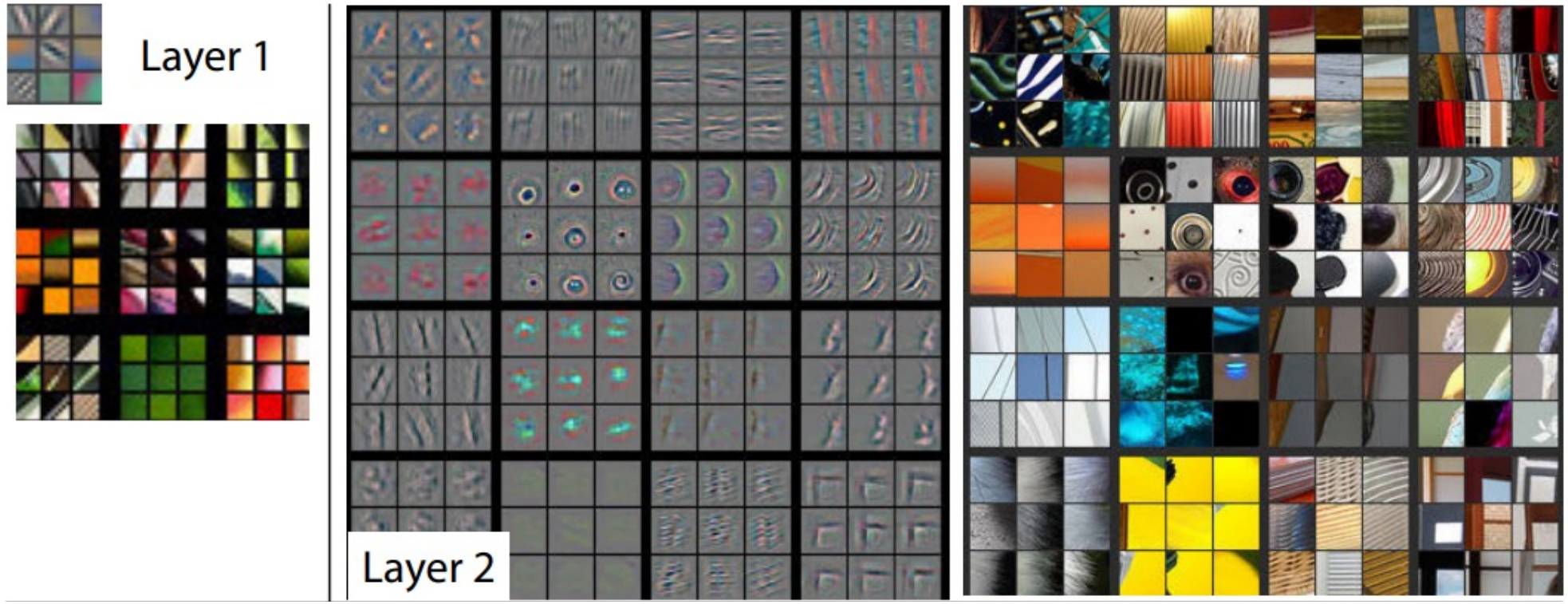
A Full Convolutional Neural Network (LeNet)

When it goes to higher level, CNN is able to capture more complicated features

CNN

Zeiler and Fergus (2014)
https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53

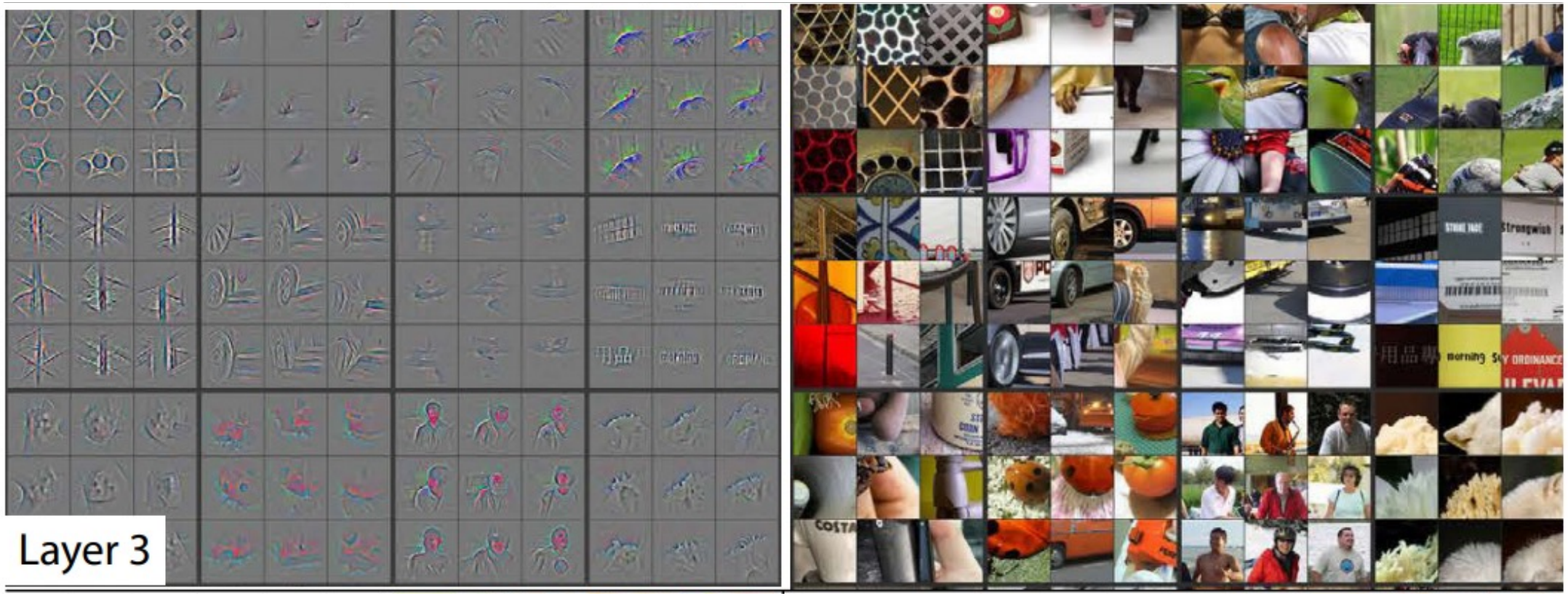
CNN captures more complicated features with deep networks



CNN

Zeiler and Fergus (2014)
https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53

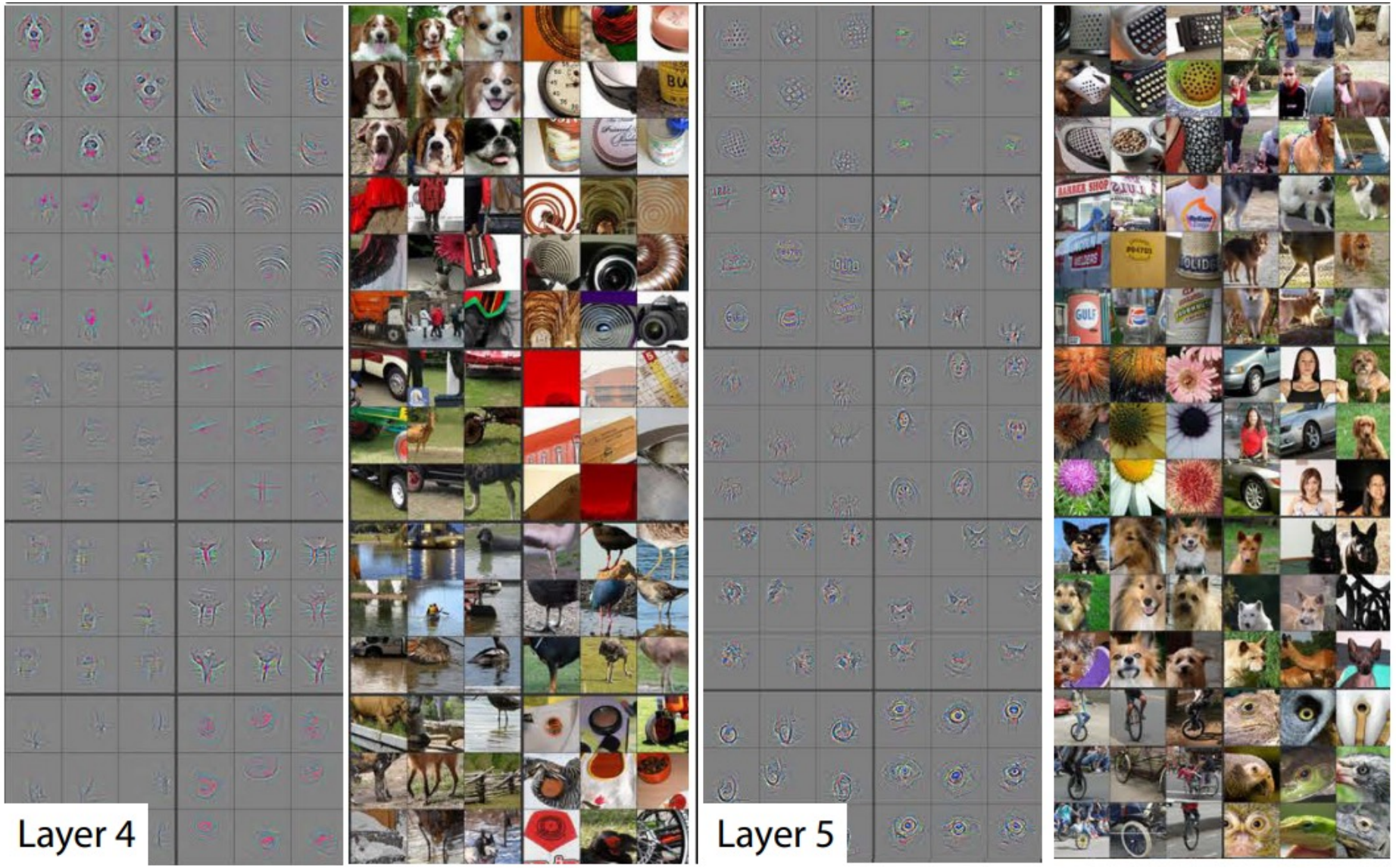
CNN captures more complicated features with deep networks



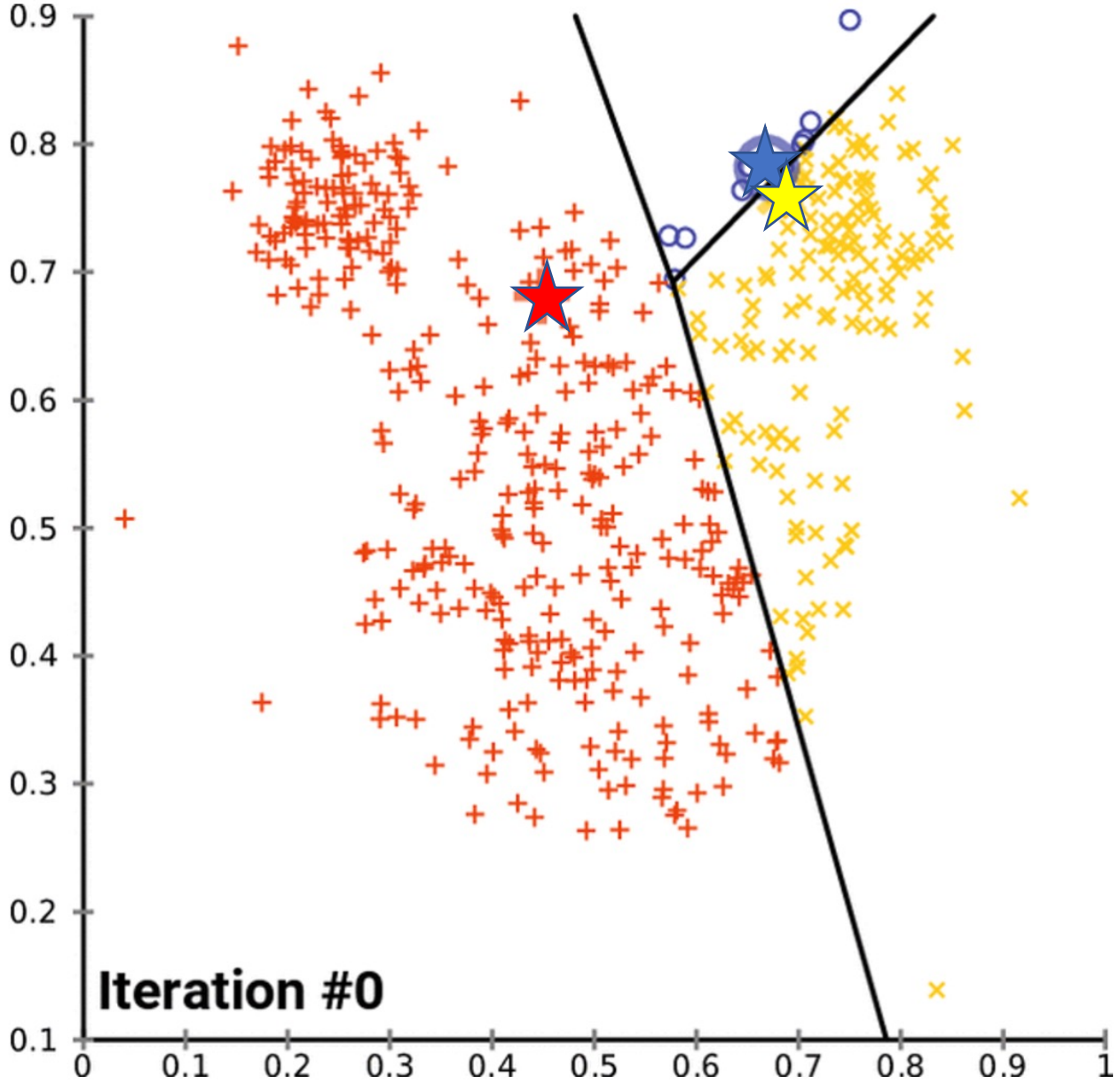
CNN

Zeiler and Fergus (2014)
https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53

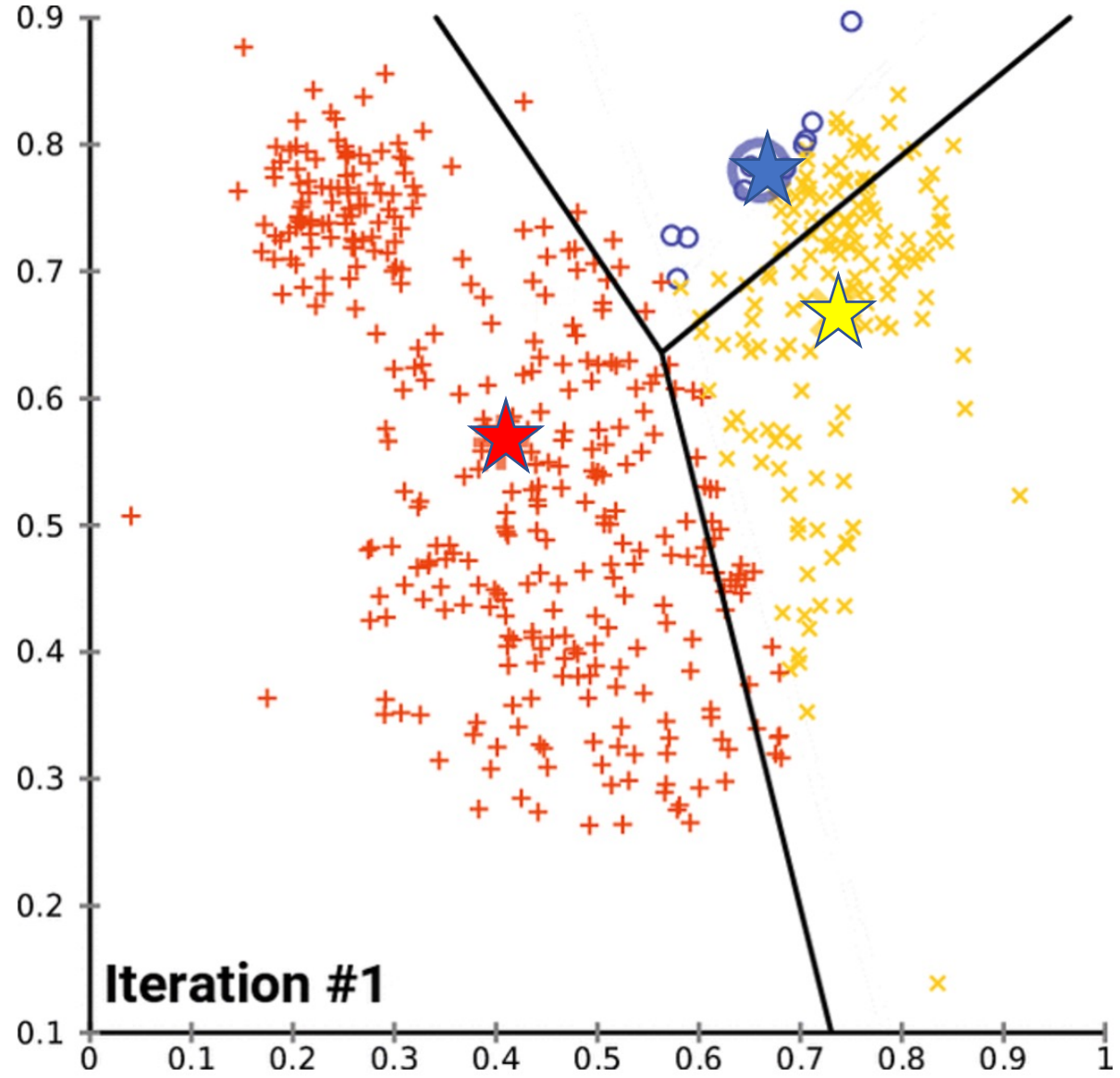
CNN captures more complicated features with deep networks



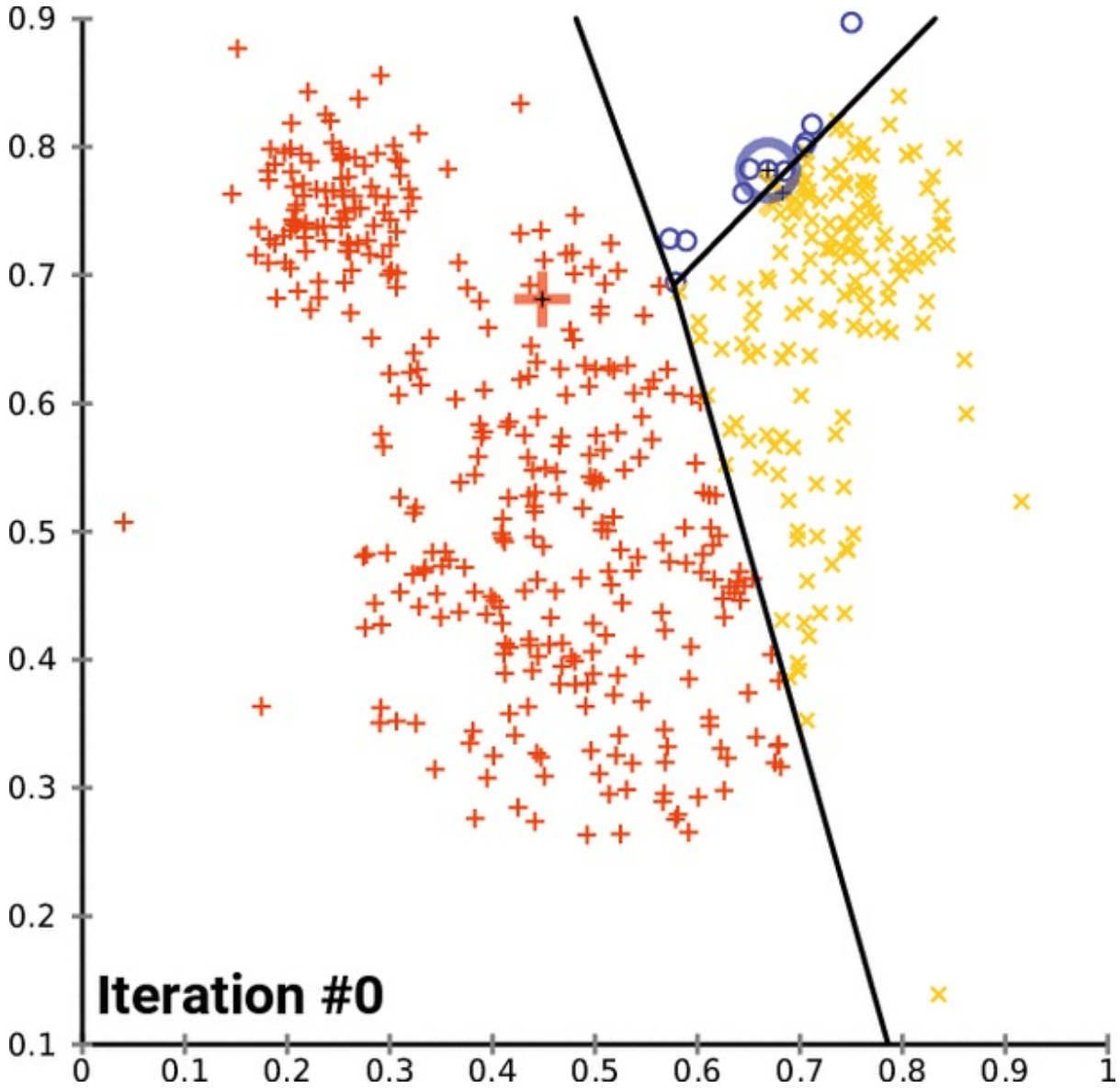
Unsupervised Learning? K-Means



Unsupervised Learning? K-Means



K-Means



Concepts of ML
ML Algorithms:
kNN, DT, RF, SVM, ANN, CNN, and K-means

Data Preprocessing Demo