Data Analytics for Materials Science

27-737

A.D. (Tony) Rollett, Elizabeth Holm,
R.A. LeSar (Iowa State Univ.)

Dept. Materials Sci. Eng., Carnegie Mellon University

Convolutional Neural Nets (CNNs)

Lecture 13

Revised: 13th Apr. 2021

Recapitulation

To date, we have discussed:

Linear algebra
Linear regression: prediction
Multiple linear regression (MLR): prediction
Regular expressions
Principal component analysis (PCA)
Canonical Correlation Analysis (CCA)
Random Forest (RF)
Clustering
Artificial Neural Nets

Resources

- Hastie et al. Elements of Statistical Learning:
  Neural Nets, starting on about p. 389
- https://victorzhou.com/blog/intro-to-neural-networks
- https://en.wikipedia.org/wiki/History_of_artificial_neural_networks
- https://scipy-lectures.org/advanced/mathematical_optimization/index.html
- http://neuralnetworksanddeeplearning.com/chap1.html
  seems to have nice simple explanations of, e.g., perceptrons, and what the adjustment of weights in the network accomplishes.
- http://cs231n.github.io/
  This appears to be a very complete set of notes on NNs, actually a complete course.
- https://machinelearningmastery.com/neural-network-models-for-combined-classification-and-regression/
- General description of NN: https://victorzhou.com/blog/intro-to-neural-networks
- For explaining forward and backward propagation: https://tech.trustpilot.com/forward-and-backward-propagation-5dc3c49c9a05

Carnegie Mellon University

Convolutional Neural Networks and Black Box AI

Elizabeth A. Holm

27-515 Introduction to Computational Materials Science

I recently watched a lecture by Geoffrey Hinton, University of Toronto and developer of CNNs.

He pointed out that it took about 30 years for the NN people to recognize that they did not have to use a logistic activation function and that a $\tanh$ function works better in some cases.

He emphasized that just because everyone uses CNNs, they are not the only way to do this problem and that new methods are likely to be easier and better.

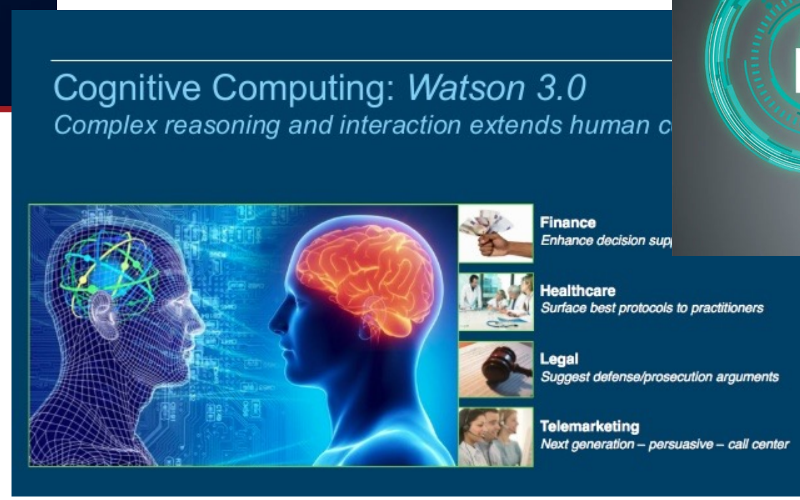
My point is that none of this is magic.

A caveat

Many of the high profile advances in computer science…

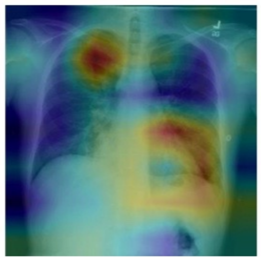
…are based on deep artificial neural networks.

Uber's First Autonomous Fleet

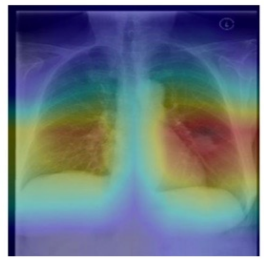


Deep learning beats humans at their own game

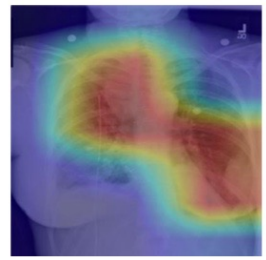
- Computer vision outperforms radiologists at detecting abnormalities in chest x-rays



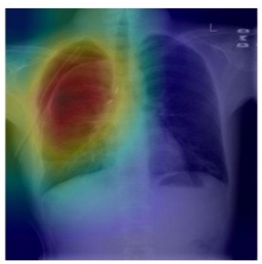
(a) Patient with multifocal community acquired pneumonia. The model correctly detects the airspace disease in the left lower and right upper lobes to arrive at the pneumonia diagnosis.



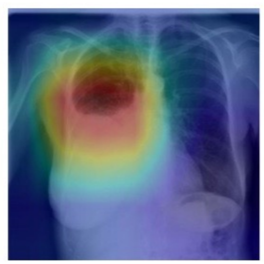
(b) Patient with a left lung nodule. The model identifies the left lower lobe lung nodule and correctly classifies the pathology.



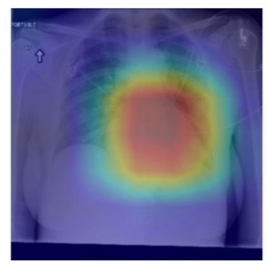
(c) Patient with primary lung malignancy and two large masses, one in the left lower lobe and one in the right upper lobe adjacent to the mediastinum. The model correctly identifies both masses in the X-ray.



(d) Patient with a right-sided pneumothroax and chest tube. The model detects the abnormal lung to correctly predict the presence of pneumothorax (collapsed lung).



(e) Patient with a large right pleural effusion (fluid in the pleural space). The model correctly labels the effusion and focuses on the right lower chest.

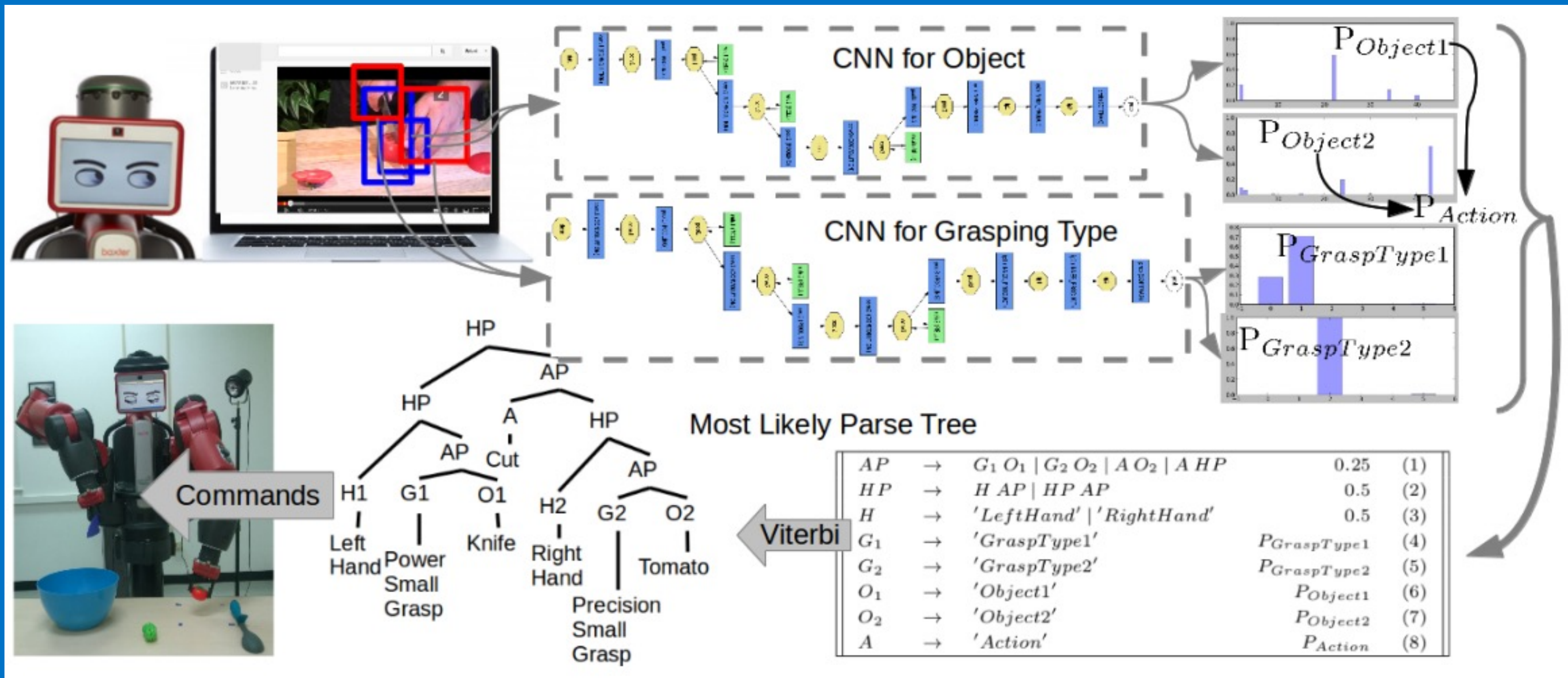


(f) Patient with congestive heart failure and cardiomegaly (enlarged heart). The model correctly identifies the enlarged cardiac silhouette.

P. Rajpurkar et al. arXiv:1711.05225

[Playing Atari with Deep Reinforcement Learning.](#)
[Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller](#)

https://e2eml.school/how_convolutional_neural_networks_work.html



[Robot Learning Manipulation Action Plans by "Watching" Unconstrained Videos from the World Wide Web. Yezhou Yang, Cornelia Fermuller, Yiannis Aloimonos]

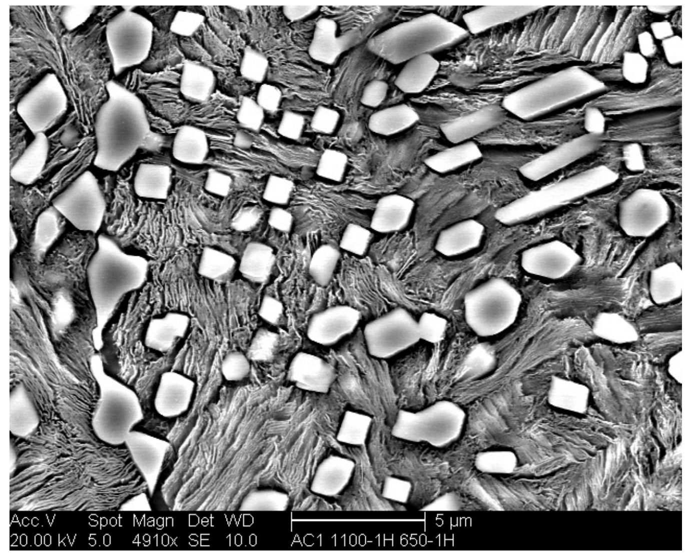
https://e2eml.school/how_convolutional_neural_networks_work.html

Let's take a moment to think about image data…

Natural Images

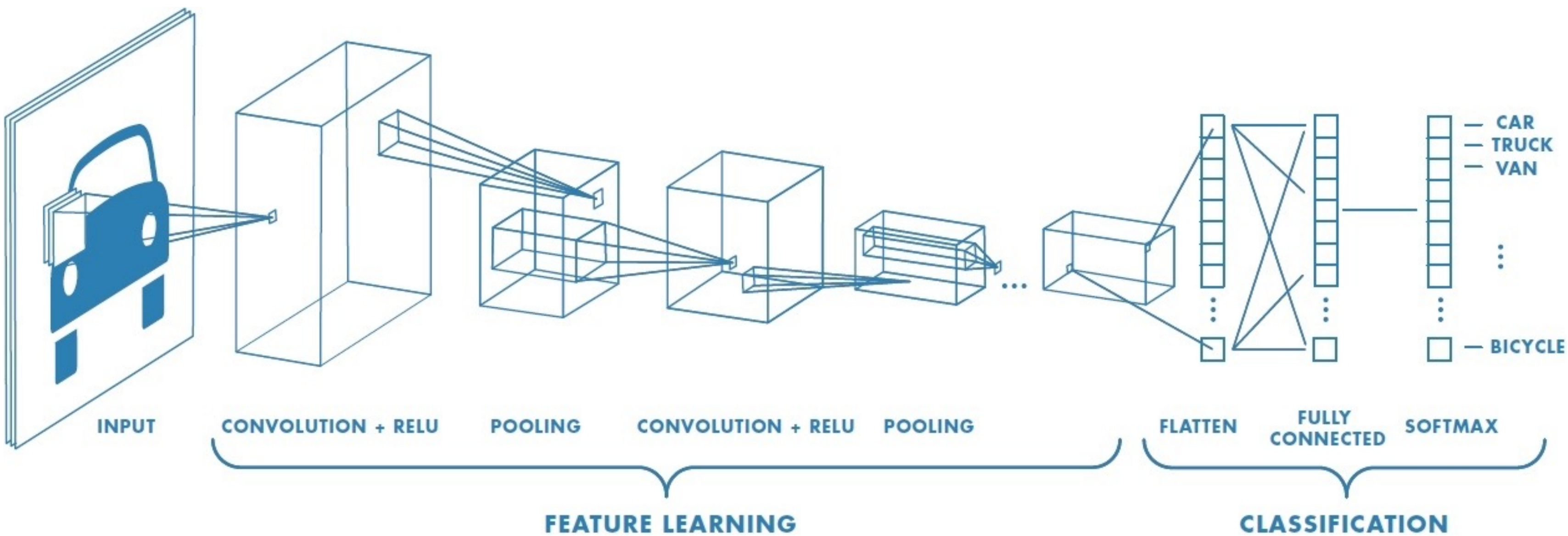
- One instance of the object
- Features at various length scales
- Arbitrary field of view
- Almost always oriented

Microstructures



- Many instances of the object(s)
- Characteristic length scale(s)
- Standard field of view
- Often not oriented

Convolutional Neural Networks for image analysis



- The **feature learning pipeline** turns the image into a string of numbers (feature vector).
- **Classification** entails the features "voting" on the image category.

Consider an X as represented pixel by pixel:

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Identify features:

1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

-1	-1	1
-1	1	-1
1	-1	-1

https://e2eml.school/how_convolutional_neural_networks_work.html

Consider an X as represented pixel by pixel:

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Identify features:

1	-1	-1
-1	1	-1
-1	-1	1

$(-1)\times 1 = -1$
$(-1)\times(-1) = 1$
$(-1)\times(-1) = 1$
$(-1)\times(-1) = -1$
$(-1)\times 1 = -1$
$(-1)\times(-1) = 1$
$1\times(-1) = -1$
$(-1)\times(-1) = 1$
$1\times 1 = 1$

$\sum/9 = 3/9 = 0.333$

Multiply each entry in the 3x3 area in the image by the equivalent entry in the filter (feature) and sum those terms and divide by the number of pixels (9):

$$\left(\boxed{(1\times 1)} + \boxed{(-1\times -1)} + \cdots\right)/9 = 1$$

This gives a measure of the correlation between the pixels in the image and that particular feature. We locate that value at the upper left corner. We do this for all pixels for all filters.

The elements of Convolutional Neural Networks: Filters

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

image

$\otimes$

1	-1	-1
-1	1	-1
-1	-1	1

filter

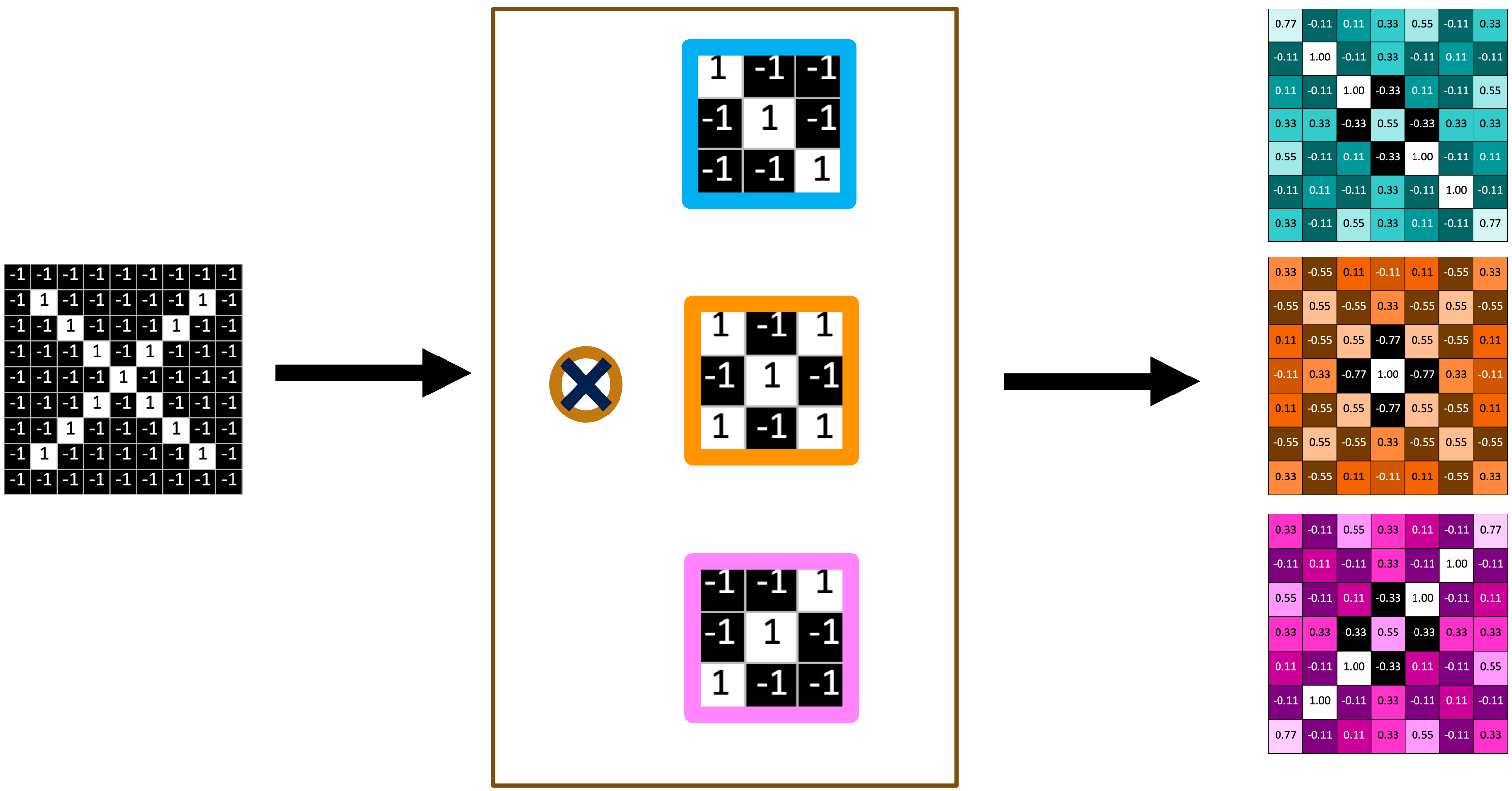
=

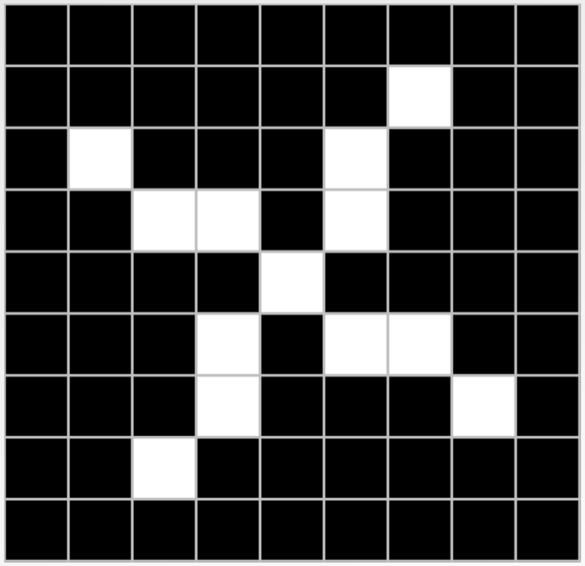
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

convolution (feature map)

https://e2eml.school/how_convolutional_neural_networks_work.html

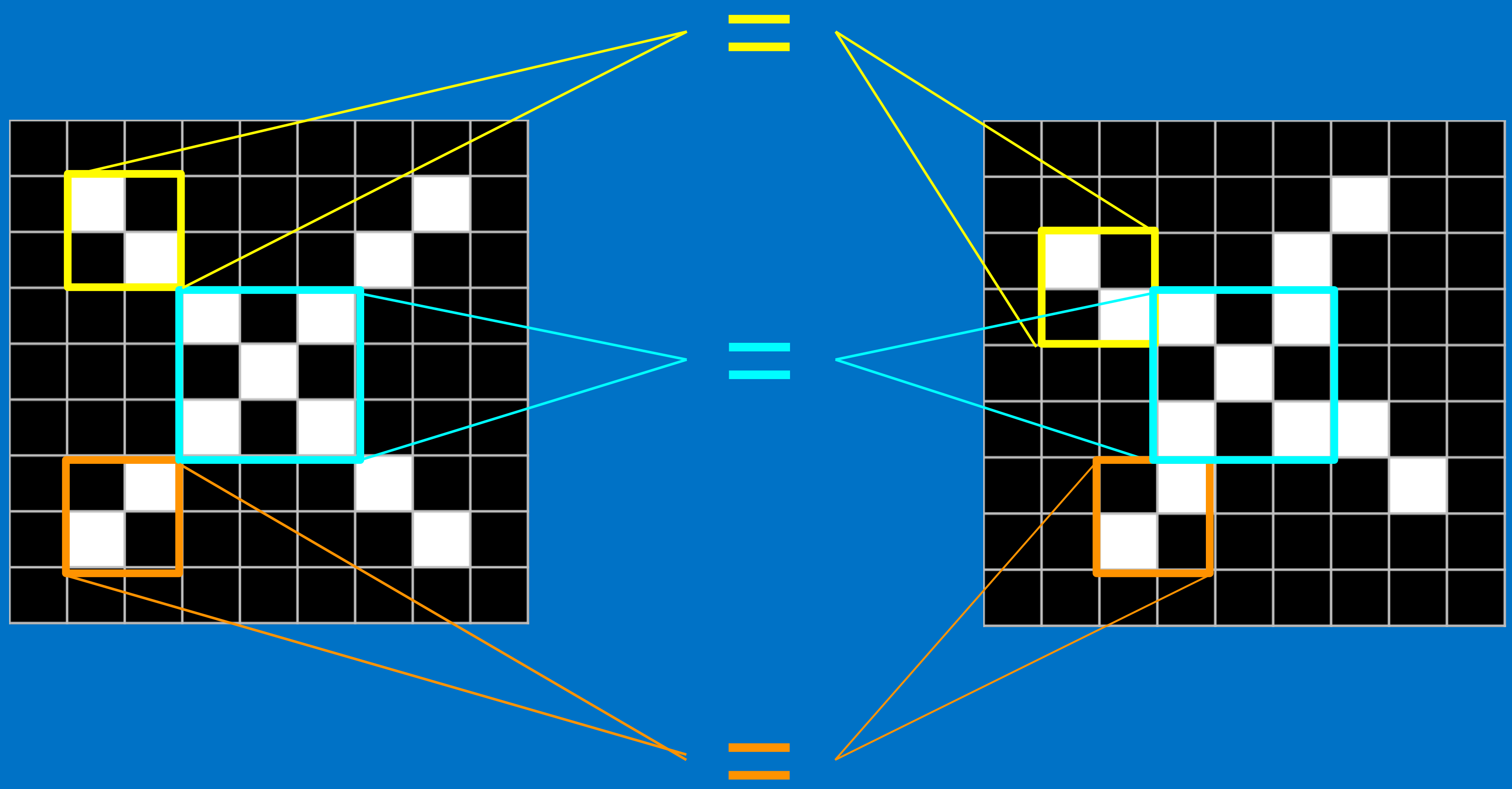
There are many filters





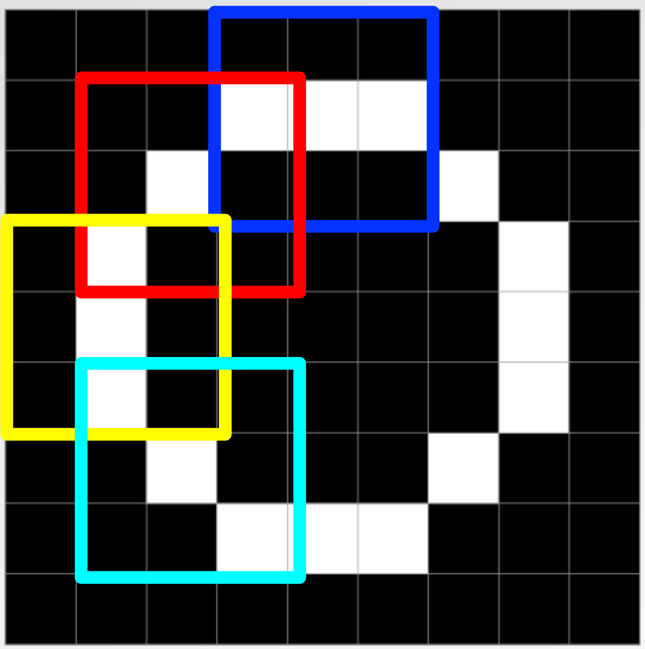
What if our X was like this?

ConvNets match pieces of the image

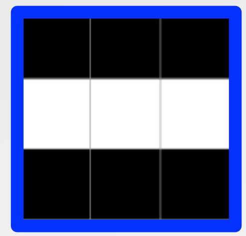


https://e2eml.school/how_convolutional_neural_networks_work.html

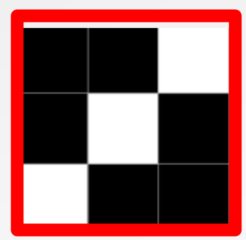
Consider an O as represented pixel by pixel:

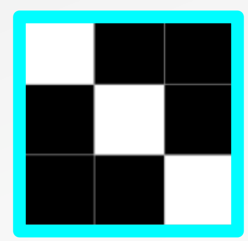


Identify features:

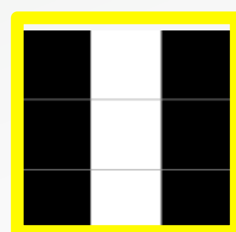


Identifies O not X





Also in X

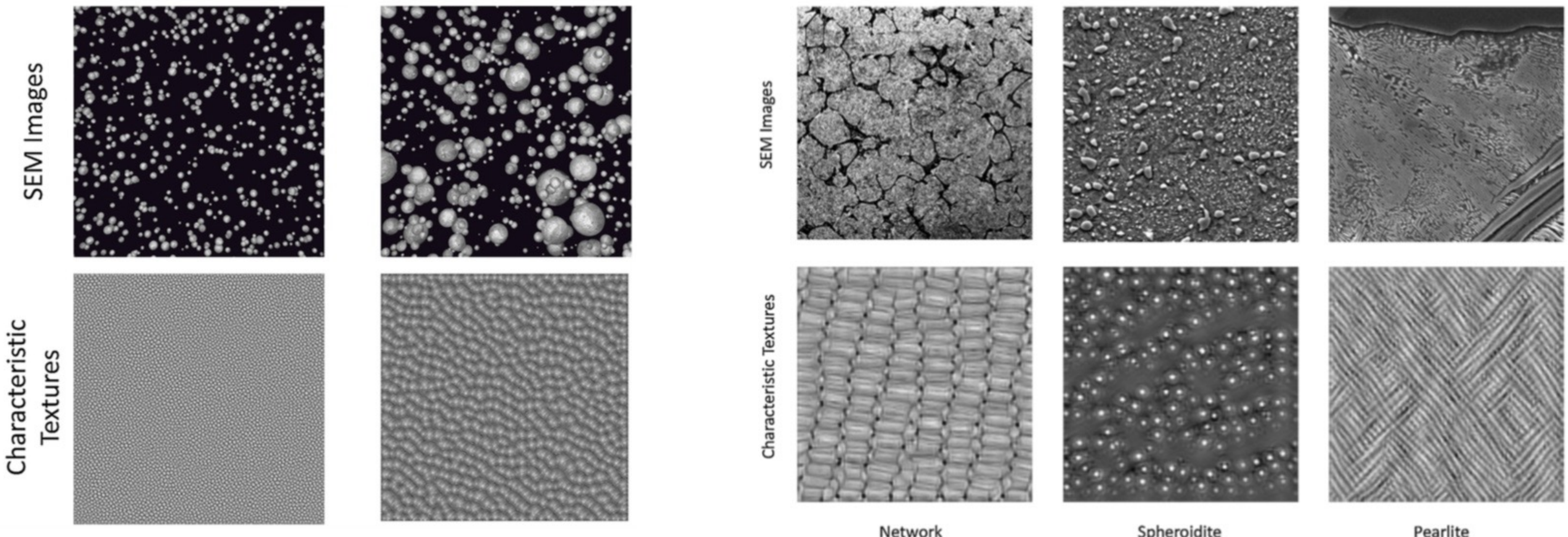


Identifies O not X

The combination of features indicate the object

Filters

Filter activations capture visual textures



- The characteristic textures make sense, and capture the features of the repeated objects.
- This is a sanity check – verification, not validation.

J. Ling et al., *Mater. Discovery* **10** 19-28 (2017)

The elements of Convolutional Neural Networks: Pooling

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

convolution

maximum

max pooling

pooling (downsampling)

1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

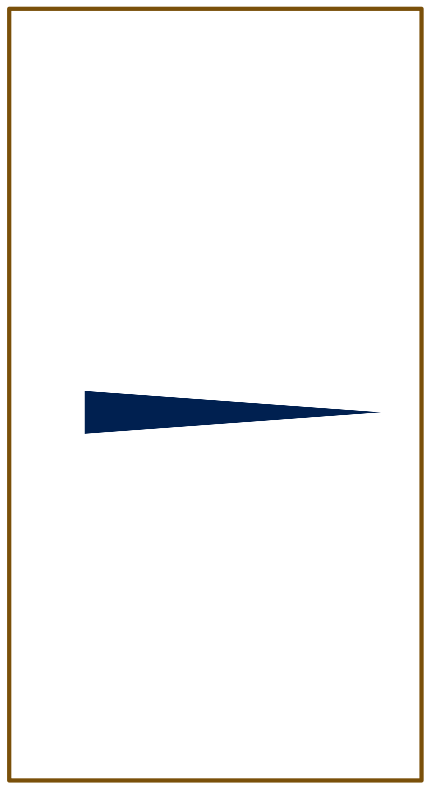
input to next convolution

Of course, the whole stack of convolutions are pooled

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.11	0.33	-0.77	1.00	-0.77	0.33	-0.11
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33

0.33	-0.11	0.55	0.33	0.11	-0.11	0.77
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33

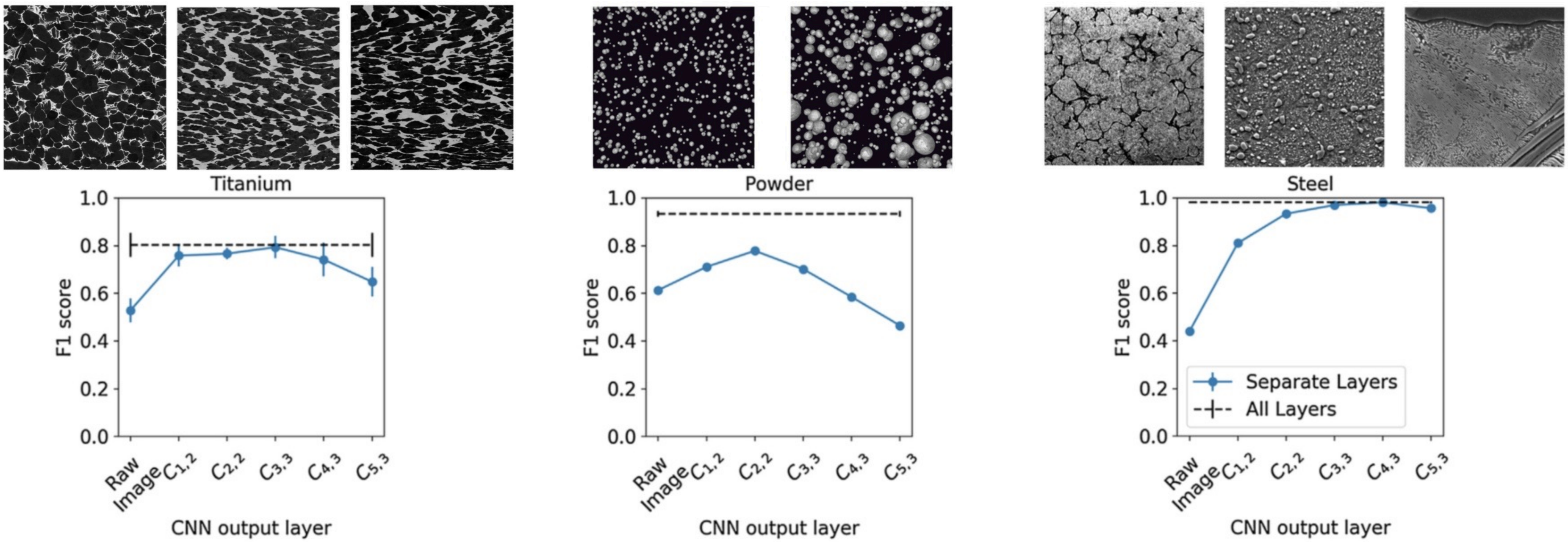


1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

0.55	0.33	0.55	0.33
0.33	1.00	0.55	0.11
0.55	0.55	0.55	0.11
0.33	0.11	0.11	0.33

0.33	0.55	1.00	0.77
0.55	0.55	1.00	0.33
1.00	1.00	0.11	0.55
0.77	0.33	0.55	0.33

Pooling captures features at different length scales



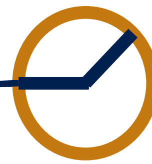
- The length scale for optimal ML corresponds to the characteristic microstructural scale.
- This length scale provides an alternative way to define a representative volume element.

J. Ling et al., *Mater. Discovery* **10** 19-28 (2017)

The elements of Convolutional Neural Networks: ReLUs

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

convolution

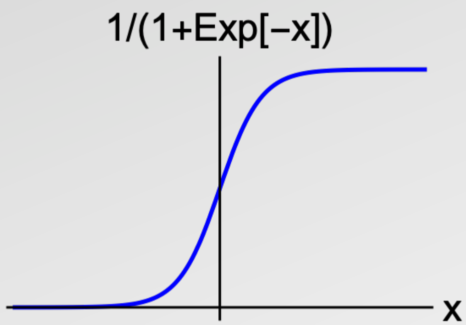


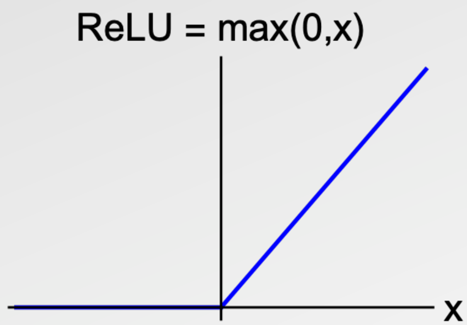
x = max(x,0)

rectifier

0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	0.77

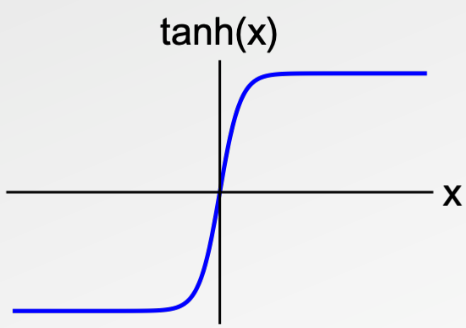
rectified convolution

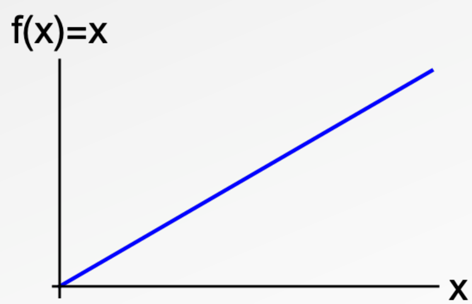




ReLU = rectified linear unit.

ReLU takes all negative values and sets them to 0. Nothing else changes.





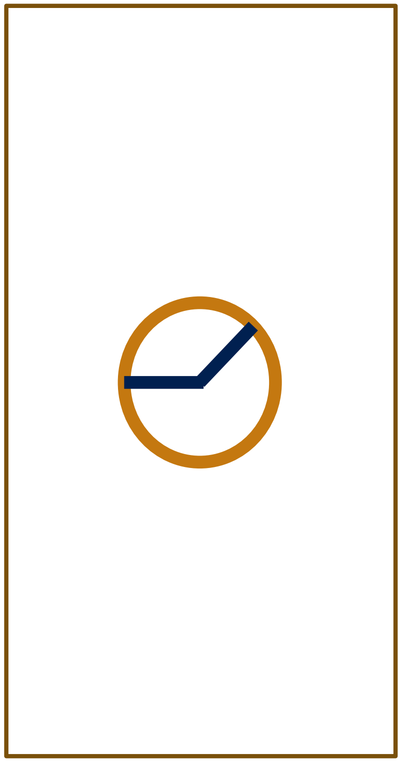
A neural net: activation functions

ReLU is applied to the full stack of convolutions, too.

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.11	0.33	-0.77	1.00	-0.77	0.33	-0.11
0.11	-0.55	0.55	-0.77	0.55	-0.55	0.11
-0.55	0.55	-0.55	0.33	-0.55	0.55	-0.55
0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33

0.33	-0.11	0.55	0.33	0.11	-0.11	0.77
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33



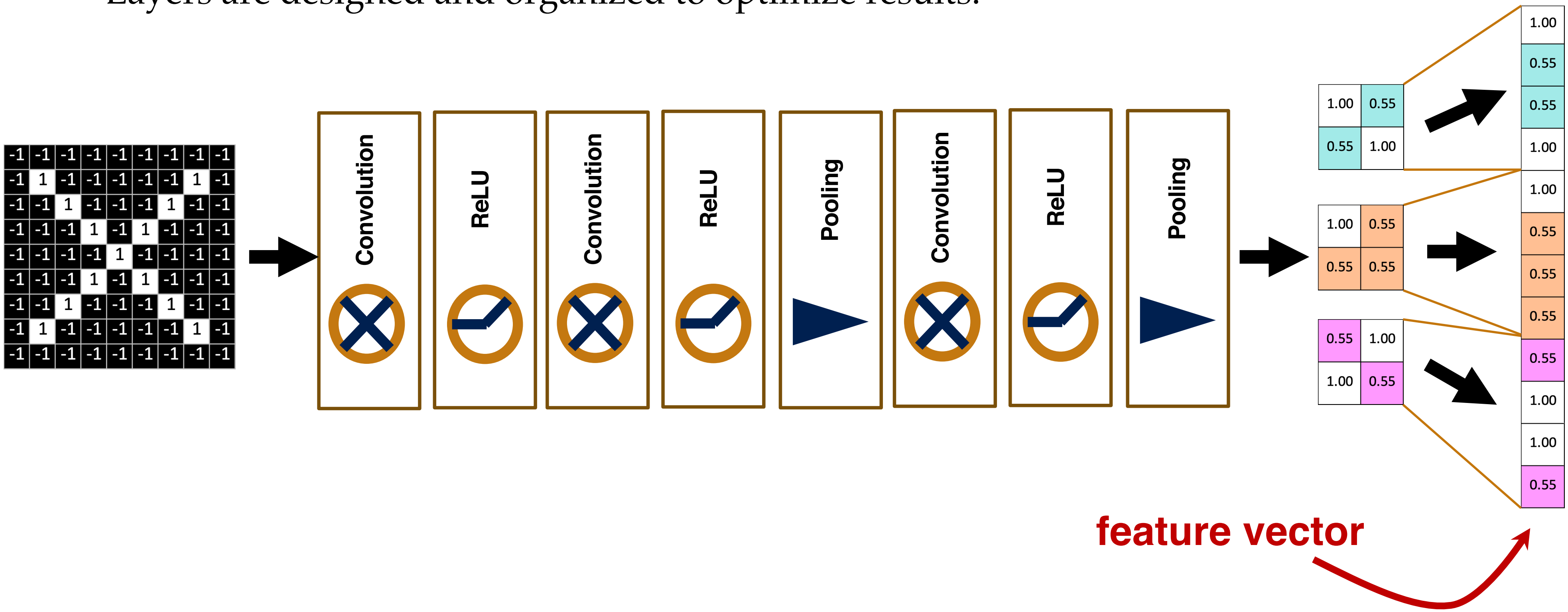
0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	0.77

0.33	0	0.11	0	0.11	0	0.33
0	0.55	0	0.33	0	0.55	0
0.11	0	0.55	0	0.55	0	0.11
0	0.33	0	1.00	0	0.33	0
0.11	0	0.55	0	0.55	0	0.11
0	0.55	0	0.33	0	0.55	0
0.33	0	0.11	0	0.11	0	0.33

0.33	0	0.55	0.33	0.11	0	0.77
0	0.11	0	0.33	0	1.00	0
0.55	0	0.11	0	1.00	0	0.11
0.33	0.33	0	0.55	0	0.33	0.33
0.11	0	1.00	0	0.11	0	0.55
0	1.00	0	0.33	0	0.11	0
0.77	0	0.11	0.33	0.55	0	0.33

Putting together the feature learning pipeline

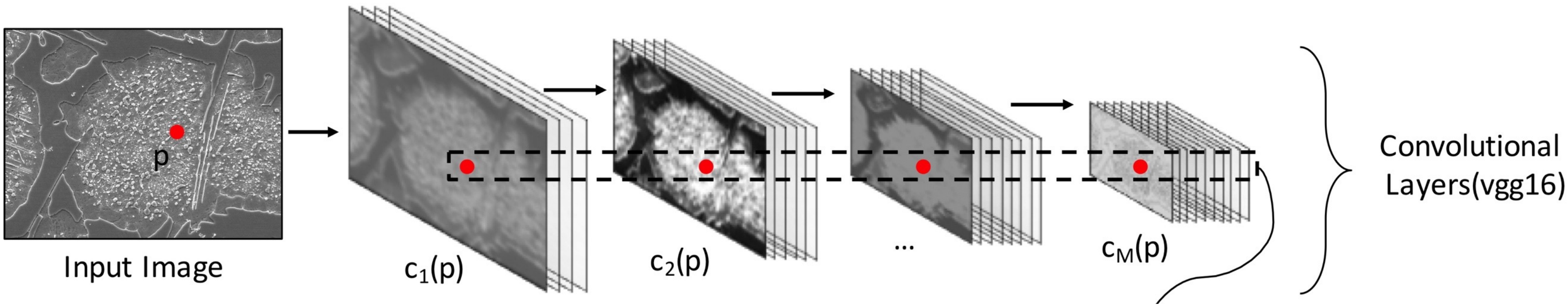
- Layers are designed and organized to optimize results.



Putting together the feature learning pipeline

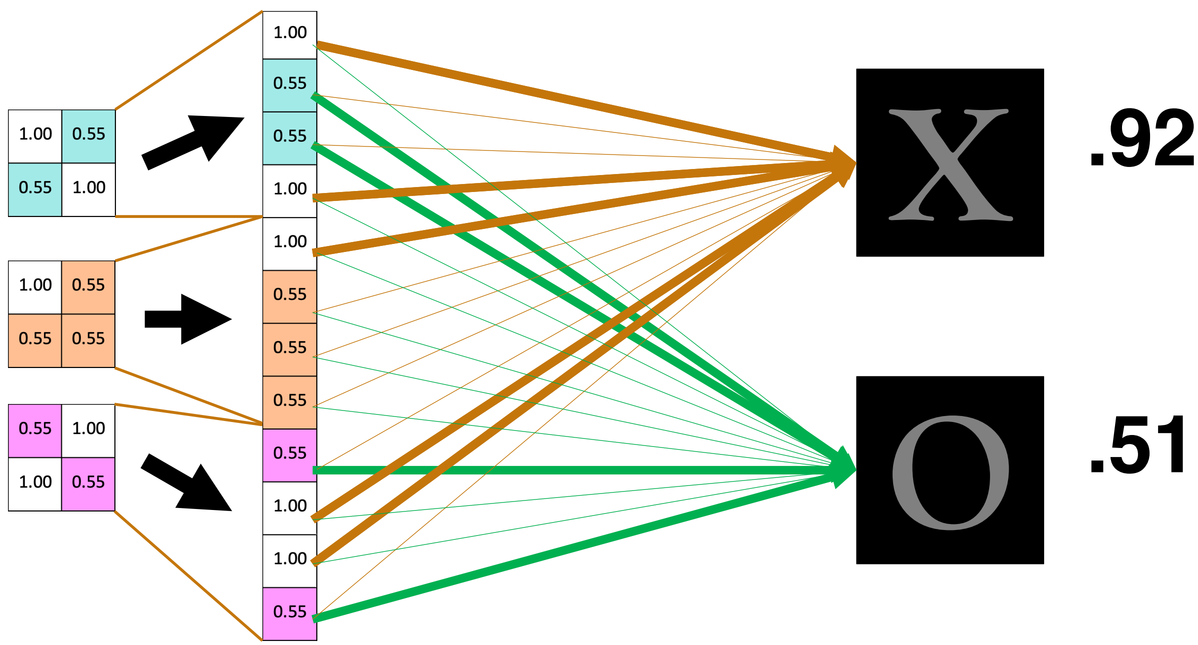
- The feature vector **numerically encodes** the visual information contained in the image.
  - Filter activations sample pixel neighborhoods
  - Pooling ensures neighborhoods are evaluated at different length scales

- Feature vectors enable **objective, autonomous microstructural analysis**.



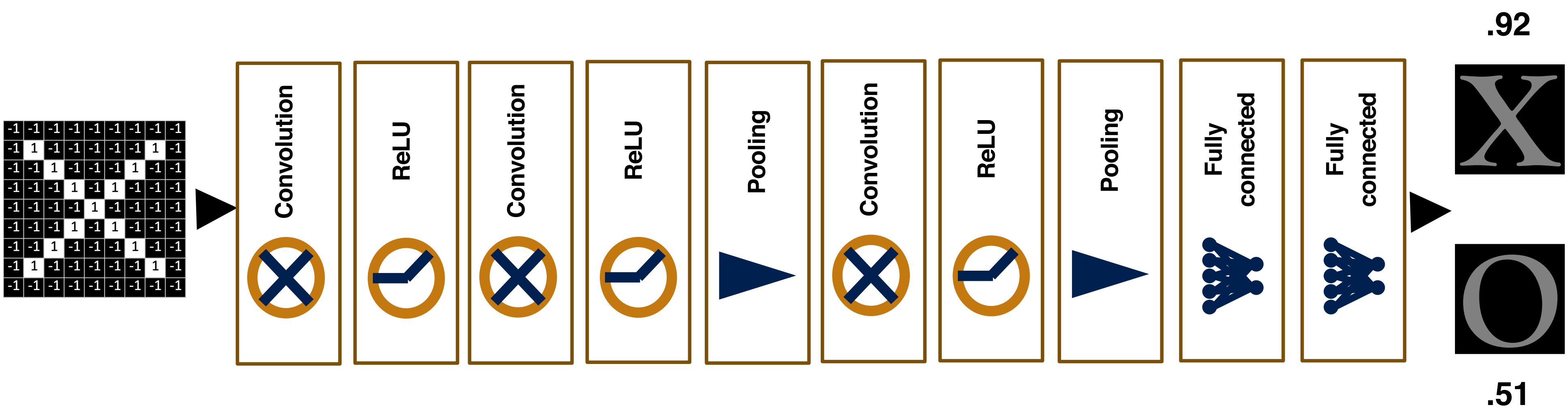
The output of Convolutional Neural Networks: Classification

- All of the final values are stacked into a list of numbers – the feature vector.
- Each entry votes to predict the outcome – some are better predictors than others!



Putting it all together

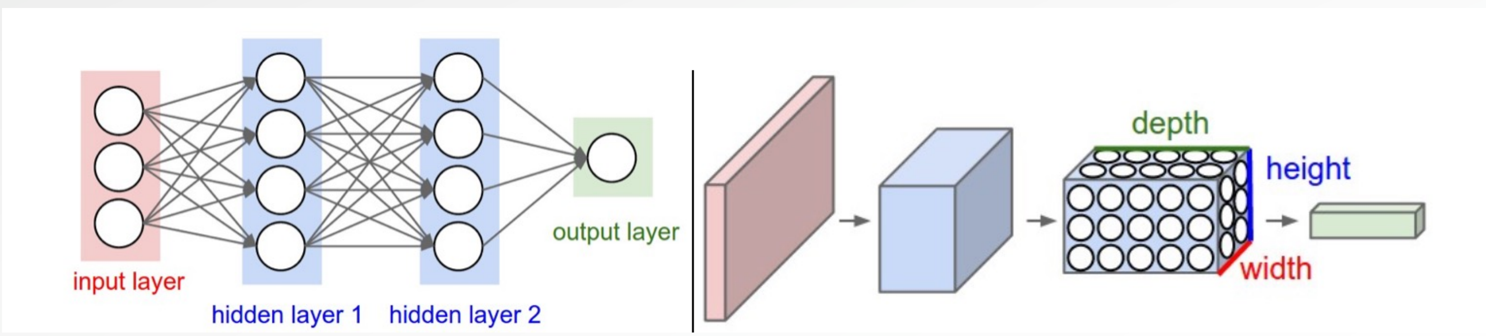
- A set of pixels becomes a set of votes.



Consider a small figure, 32 pixels by 32 pixels and 3 color channels. A single fully connected neuron in a first hidden layer in a regular neural net would have $32\times32\times3 = 3072$ weights. Scaling to an a more reasonably sized image, such as $200x200x3$ would have 120,000 weights:

- a huge number of parameters that would quickly lead to *overfitting.*

In CNNs, input images are taken as a volume, e.g., a $32x32x3$ image, and the neurons in a layer will only be connected to small volume in the previous layer. The output layer is reduced to dimensions (in this case) of $1x1x10$ (the feature vector).

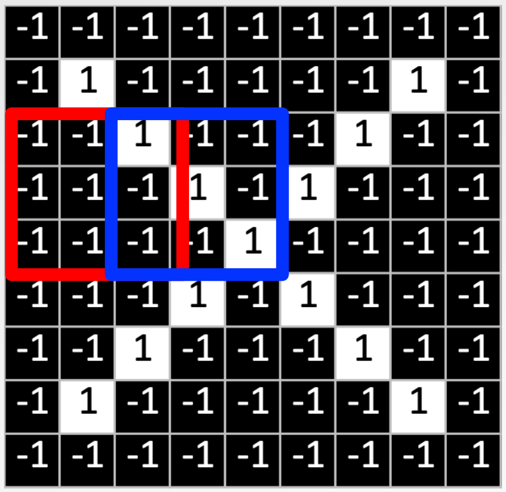


Why CNNs not NNs?

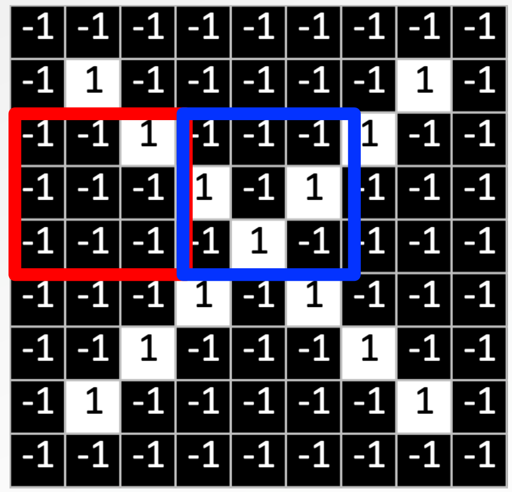
[Notes from Stanford CS 231 course]

Convolution in action: https://cs231n.github.io/convolutional-networks/#fc

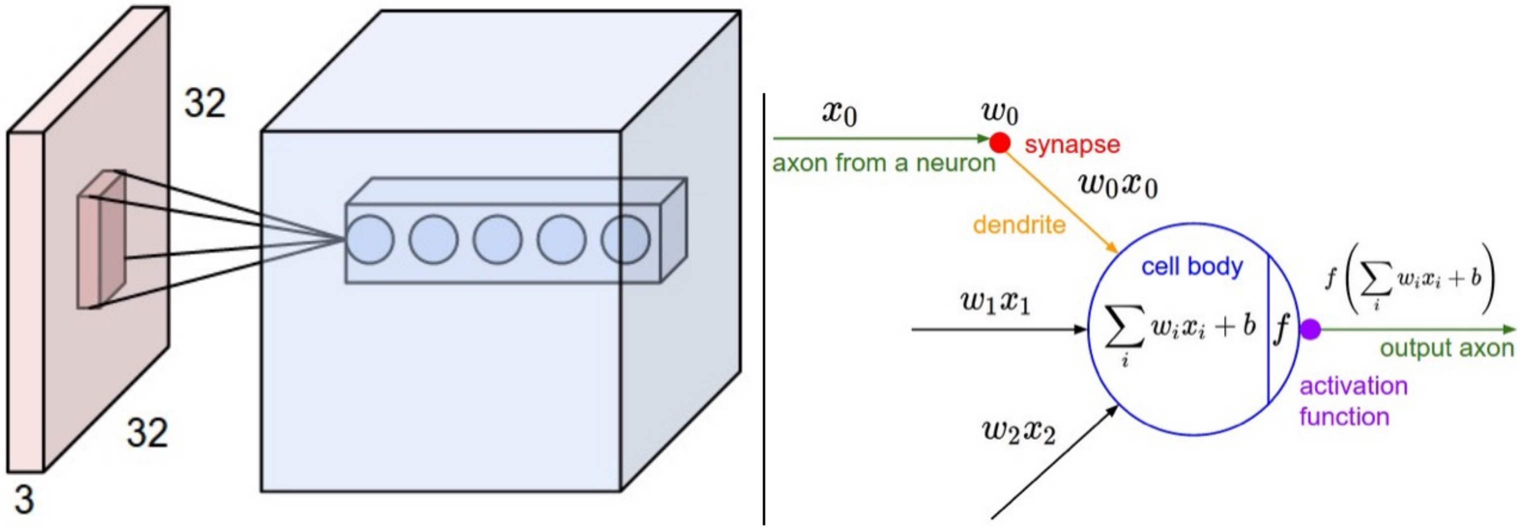
- Convolution is essentially a dot product between the filters and local regions of the input.
- Note that convolution depends on the size of the area and the stride, which controls the dimensions after convolution.



stride of 2



stride of 3

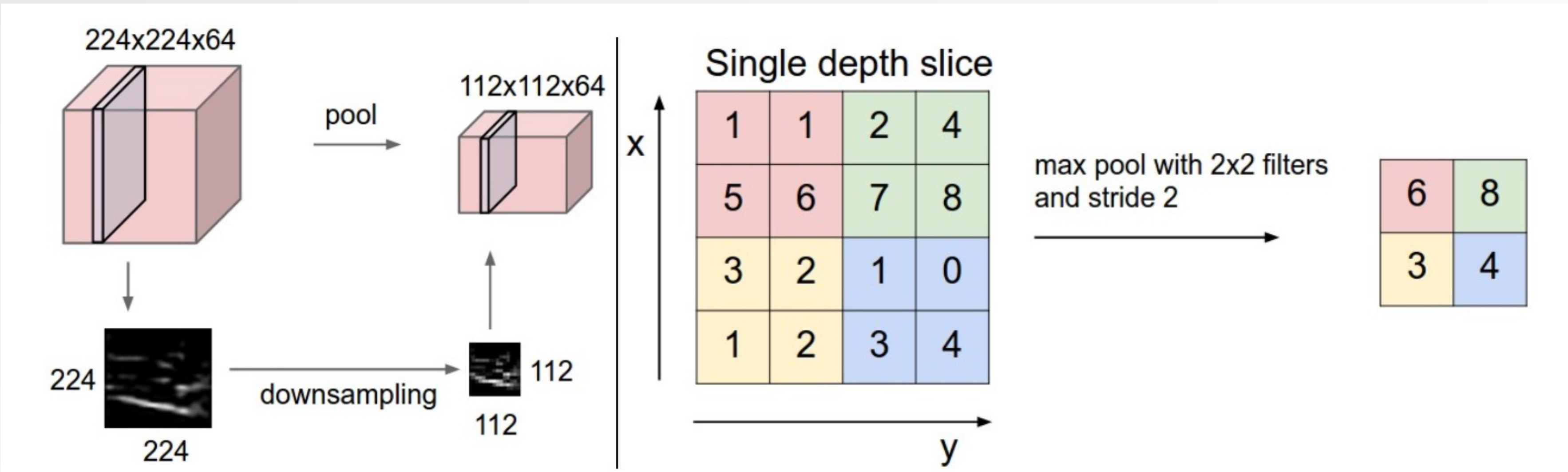


Left: An example input volume in red (e.g. a 32x32x3 CIFAR-10 image), and an example volume of neurons in the first Convolutional layer. Each neuron in the convolutional layer is connected only to a local region in the input volume spatially, but to the full depth (i.e. all color channels). Note, there are multiple neurons (5 in this example) along the depth, all looking at the same region in the input - see discussion of depth columns in text below. **Right:** The neurons from the Neural Network chapter remain unchanged: They still compute a dot product of their weights with the input followed by a non-linearity, but their connectivity is now restricted to be local spatially.

Convolution as a neural net

[Notes from Stanford CS 231 course]

"Inserting a Pooling layer in-between successive Conv layers in a ConvNet architecture progressively reduces the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice and resizes it using the MAX operation. The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 downsamples every depth slice in the input by 2 along both width and height, discarding 75% of the activations." (edited for brevity)



Pooling

[Notes from Stanford CS 231 course]

Some have suggested that one could drop the pooling stage and replace it with convolutions with a larger stride.

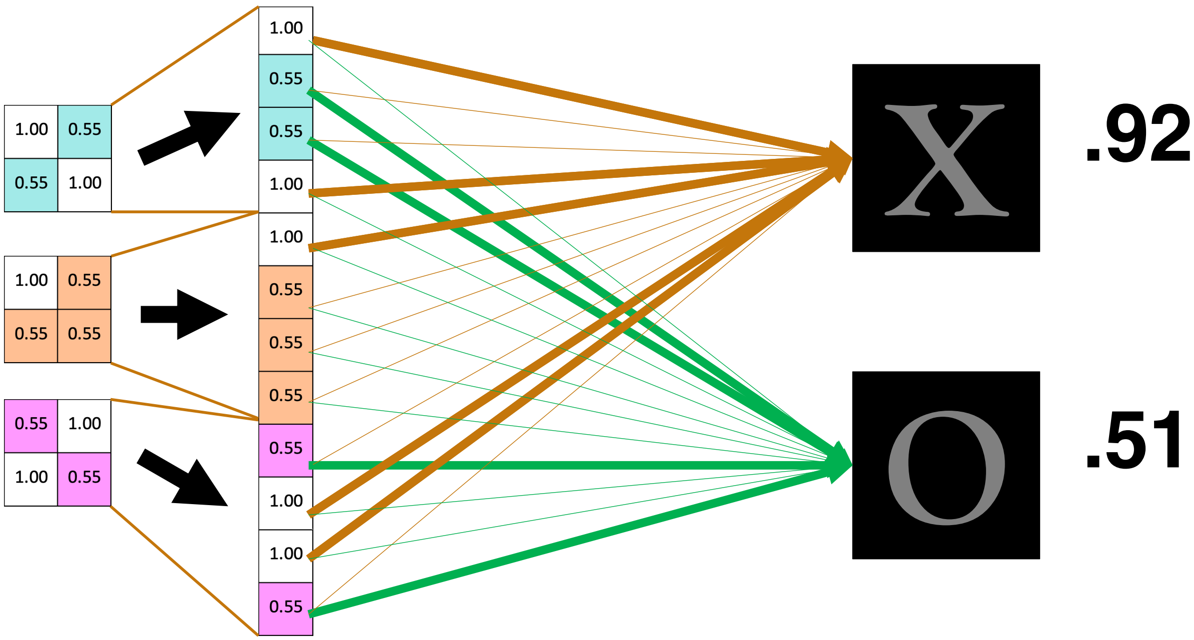
That would still reduce the size (and number of parameters)

https://arxiv.org/pdf/1412.6806.pdf

Pooling or strictly Convolution?

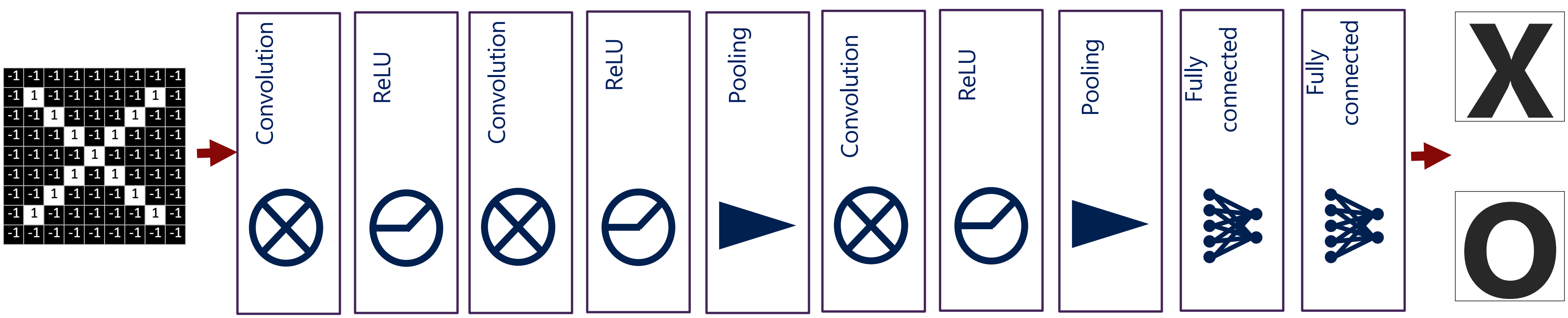
The output of Convolutional Neural Networks: Classification

- All of the final values are stacked into a list of numbers – the feature vector.
- Each entry votes to predict the outcome – some are better predictors than others!



Backpropagation

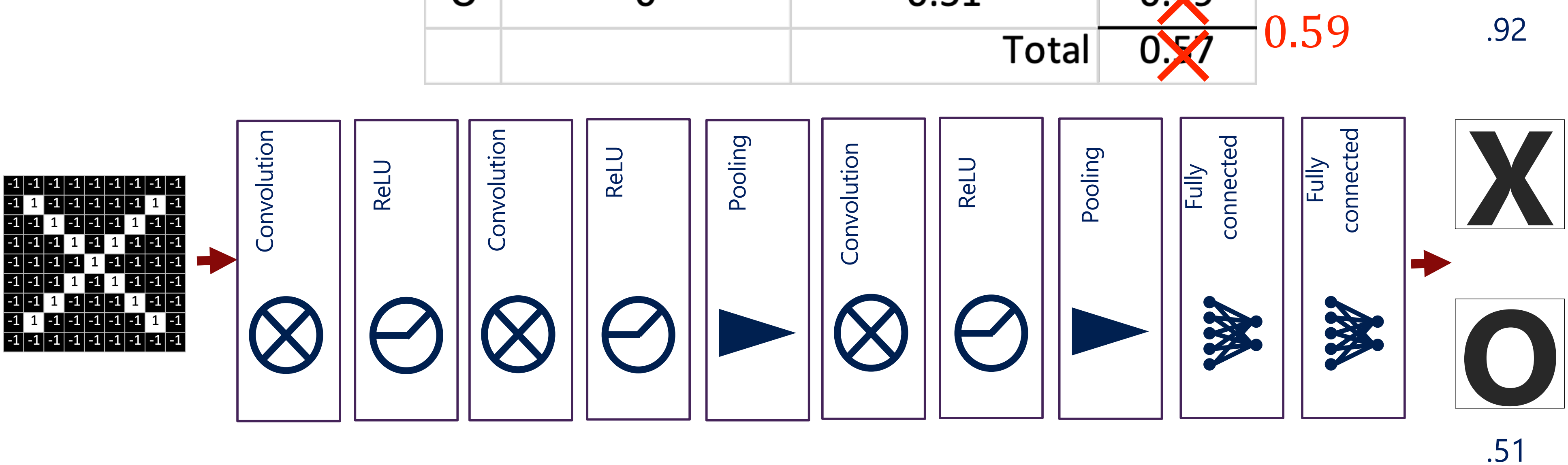
Error = right answer – actual answer



https://e2eml.school/how_convolutional_neural_networks_work.html

Backpropagation

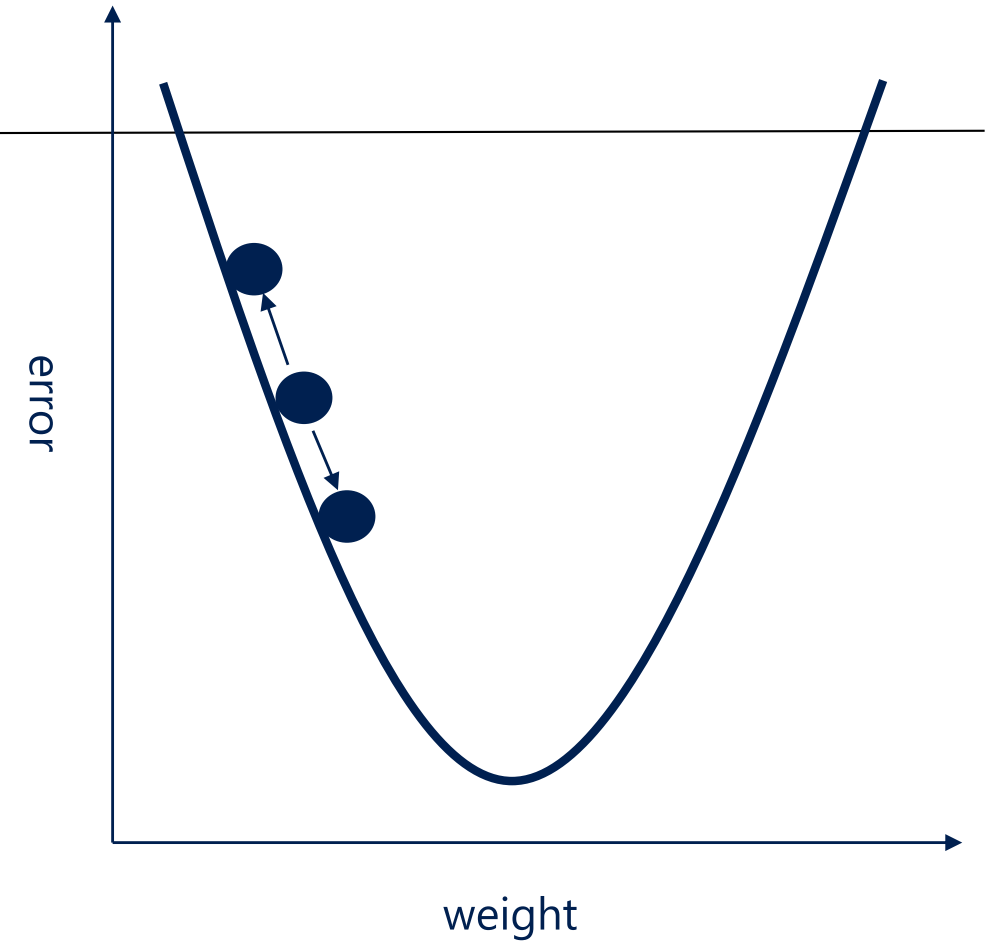
	Right answer	Actual answer	Error
X	1	0.92	0.08
O	0	0.51	~~0.49~~ 0.51
		Total	~~0.57~~ 0.59



https://e2eml.school/how_convolutional_neural_networks_work.html

Gradient descent

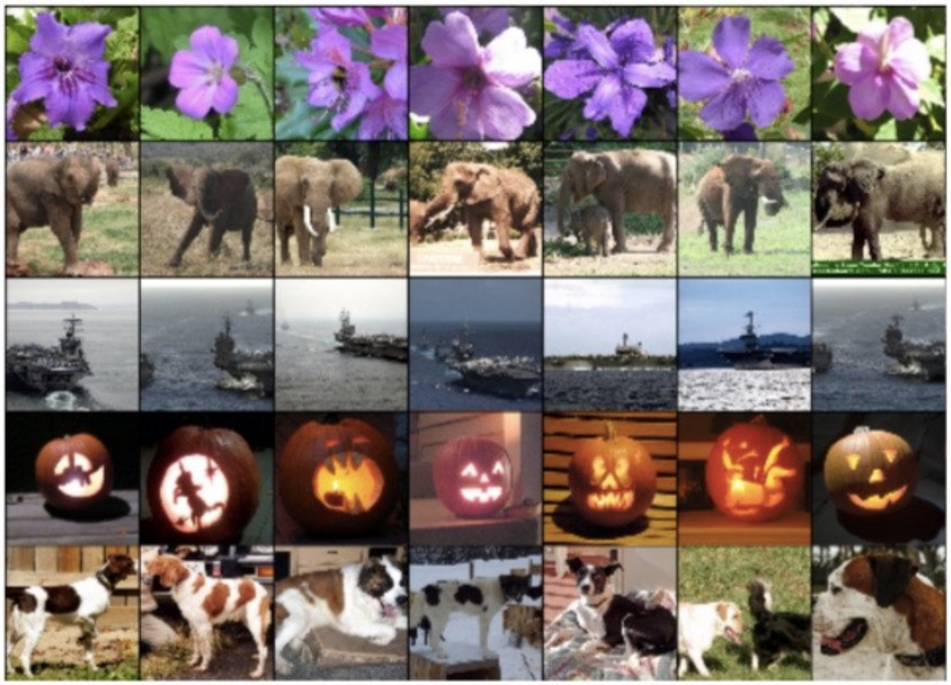
For each feature pixel and voting weight, adjust it up and down a bit and see how the error changes.



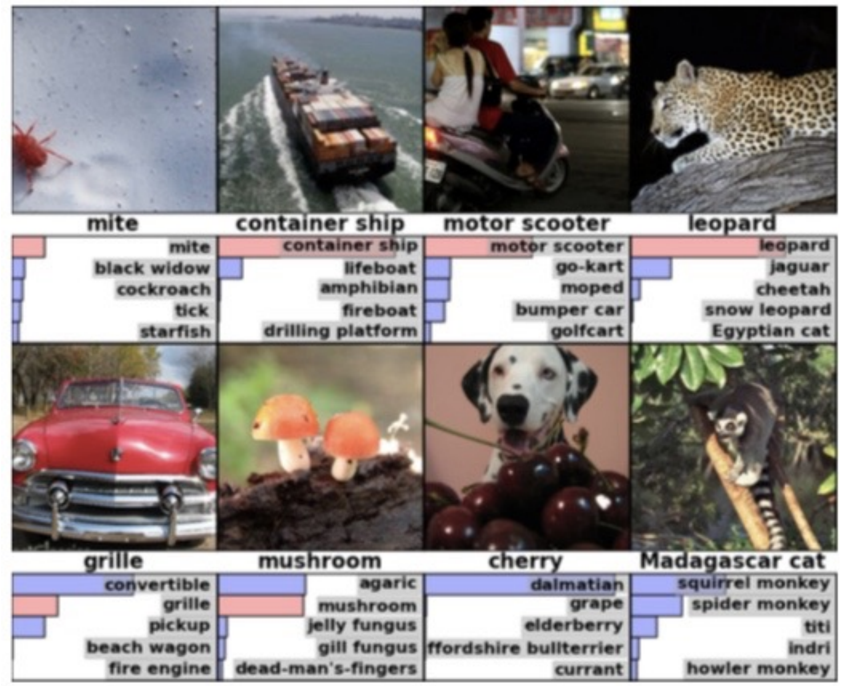
https://e2eml.school/how_convolutional_neural_networks_work.html

Convolutional Neural Networks for image analysis: Results

Use the feature vector to compute visual similarity:



Use voting to perform classification:



Krizhevsky, Alex & Sutskever, Ilya & E. Hinton, Geoffrey. (2012). Neural Information Processing Systems. 25. 10.1145/3065386.

Hyperparameters (knobs)

Convolution

- Number of features
- Size of features
- Stride

Pooling

- Window size
- Window stride

Fully Connected

- Number of neurons

https://e2eml.school/how_convolutional_neural_networks_work.html

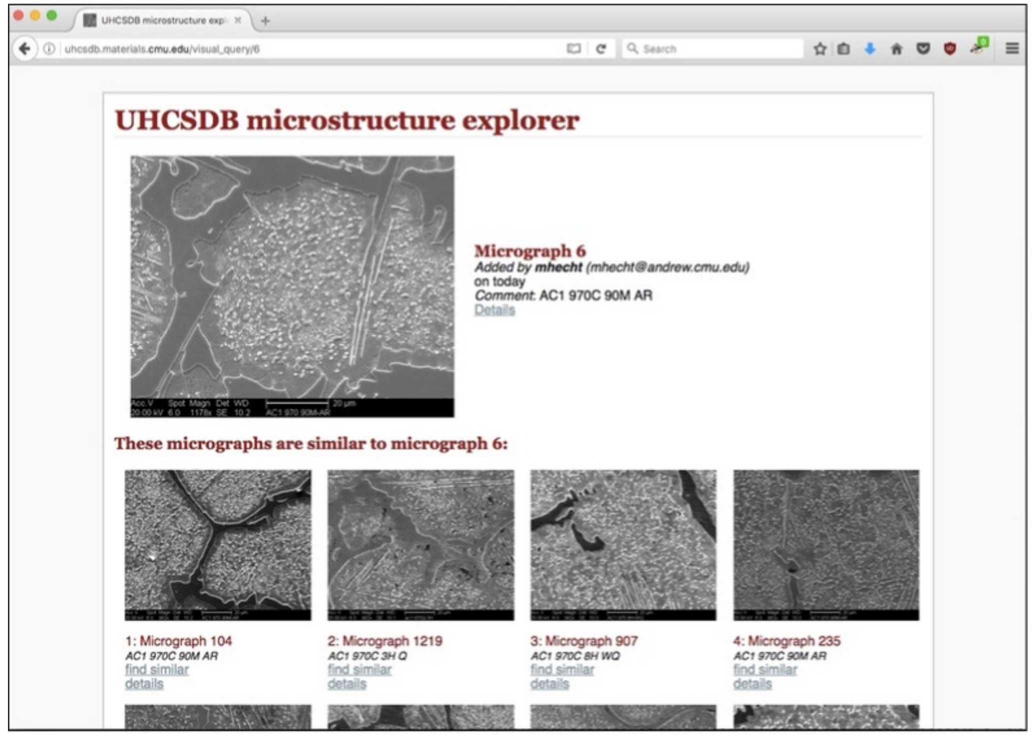
Architecture

How many of each type of layer?

In what order?

Application to microstructural science: Visual similarity

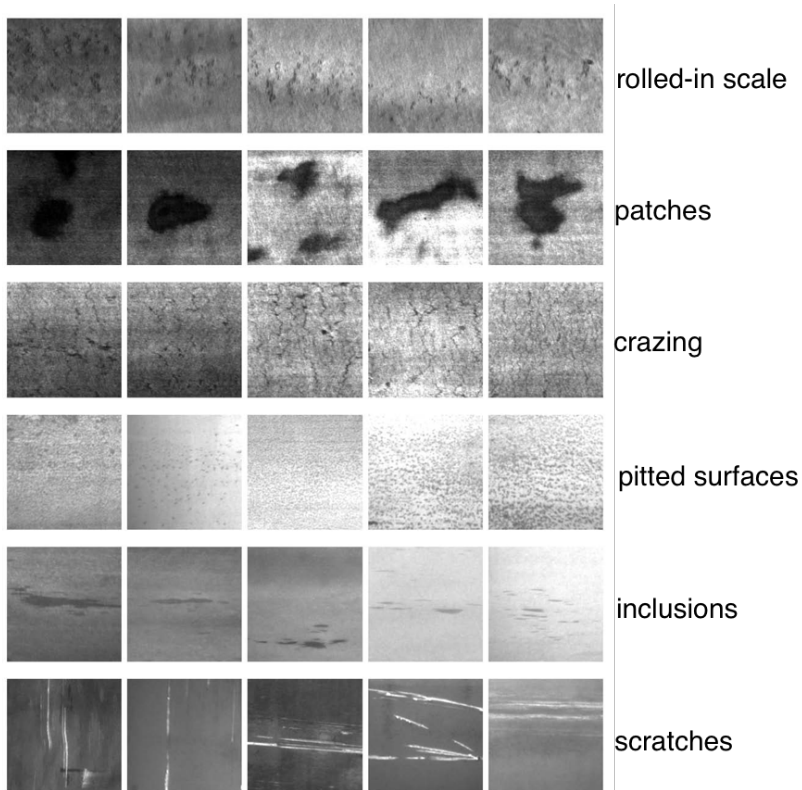
- The feature vector can be used to compare images on the basis of visual content.
- Computing visual similarity naturally translates to a visual search engine.
- Access corporate knowledge permanently and efficiently: http://uhcsdb.materials.cmu.edu/
  - *Institutional memory independent of individual memory.*

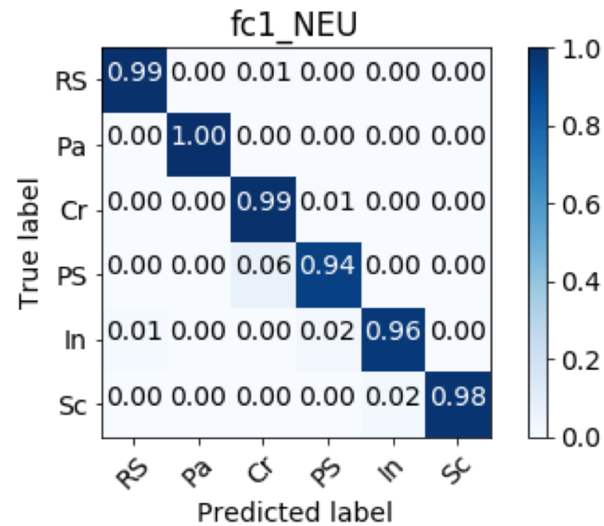


B. L. DeCost, et al., *IMMI* **6** 197-205 (2017)

Application to microstructural science: Classification

- 1800 images of surface defects in hot-rolled steel
- Use feature vector with machine learning to classify by defect type
- The "NEU" standard problem





98.3% accuracy

A. R. Kitahara, et al., *IMMI* **7** 148-156 (2018)

Biases: AIs can be misinterpreted

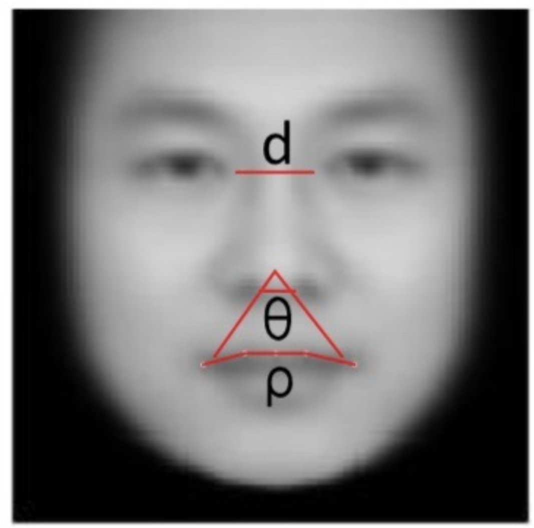
- An AI system identified “criminal” vs. “law abiding” faces with 89.5 % accuracy.

The data set:

(a) Three samples in criminal ID photo set S_c.

(b) Three samples in non-criminal ID photo set S_n

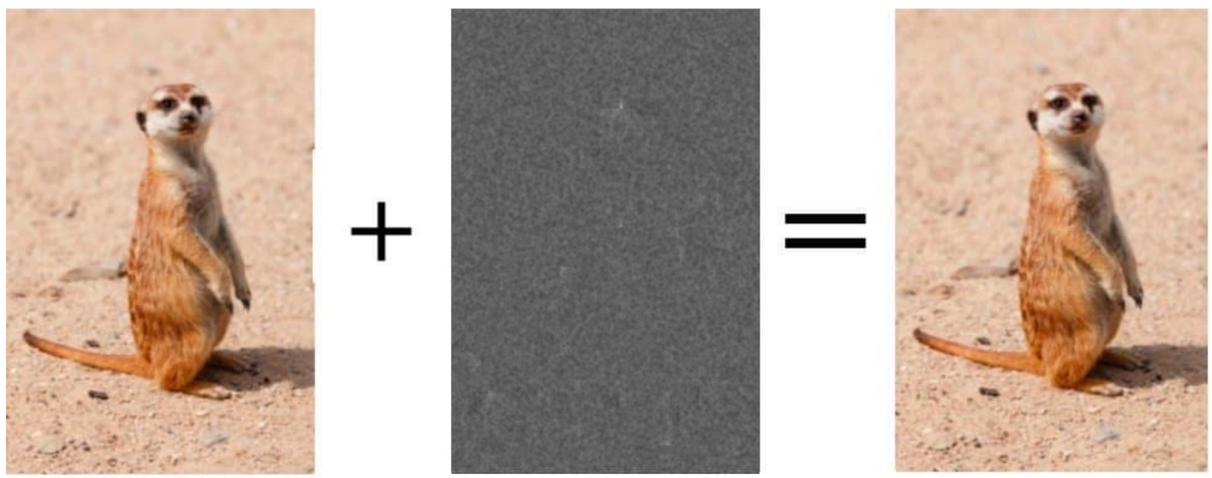
Interpreting the differences:



X. Wu and X. Zhang, arXiv 1611.04135v2

Brittleness: AIs can be fooled

- Adversarial images can cause a CNN to make spectacularly wrong decisions.



meerkat, mierkat (score = 0.90021)
mongoose (score = 0.02666)
Windsor tie (score = 0.00072)
otter (score = 0.00069)
doormat, welcome mat (score = 0.00055)

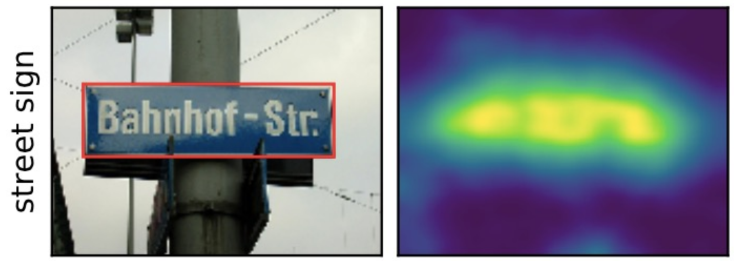
kite (score = 0.07896)
bald eagle, American eagle, Haliaeetus leucocephalus (score = 0.04153)
bee eater (score = 0.03940)
parachute, chute (score = 0.02724)
hummingbird (score = 0.02334)

doormat, welcome mat (score = 1.00000)

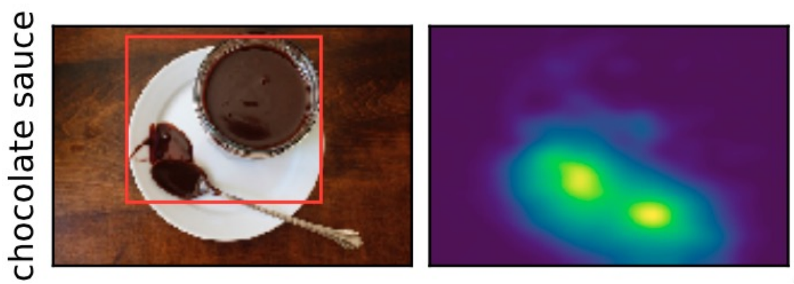
https://cv-tricks.com/how-to/breaking-deep-learning-with-adversarial-examples-using-tensorflow/

Fallability: AIs don't always learn the right things

- A CNN-based deep learning system was trained to identify classes of objects in photographs.
- Masking was used to evaluate critical features that the computer associates with an object.
- Some masks made sense:



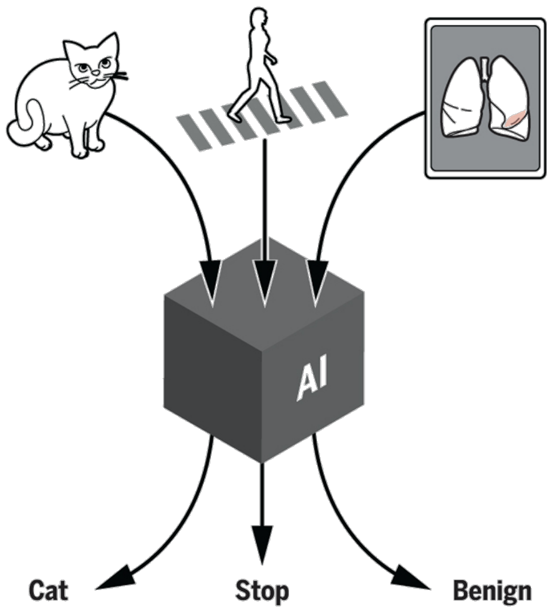
- Some did not:



Fong et al., arXiv:1704.03296v1

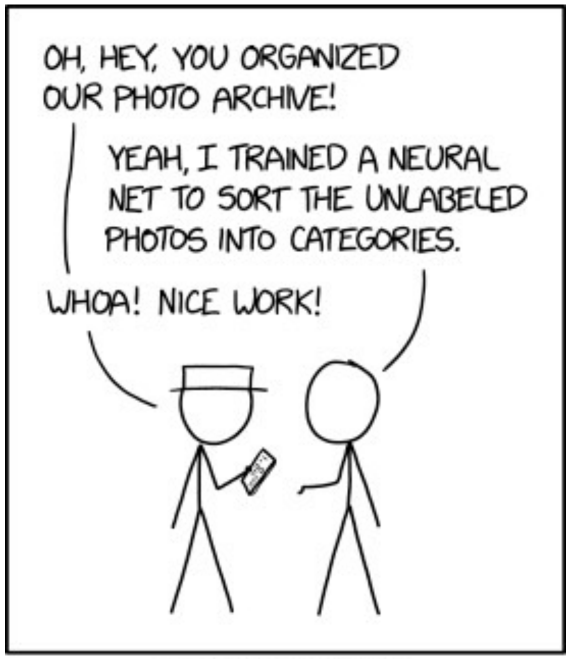
Interpreting AI answers

- There are two modes of artificial intelligence:
  - Interpretable = Basis for decision is known
  - Black Box = Basis for decision is unknown
- Scientists naturally tend to avoid black box models.



E. A. Holm *Science* 364:6435 26-27 (2019)

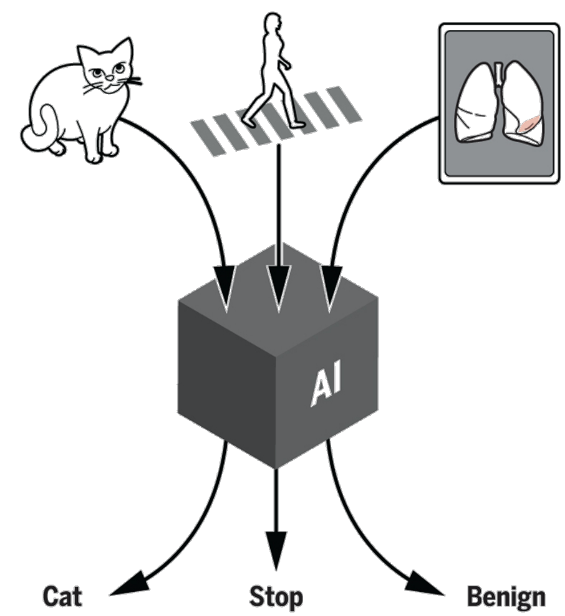
And yet … we use black box intelligence all the time



ENGINEERING TIP:
WHEN YOU DO A TASK BY HAND, YOU CAN TECHNICALLY SAY YOU TRAINED A NEURAL NET TO DO IT.

Embracing the black box…

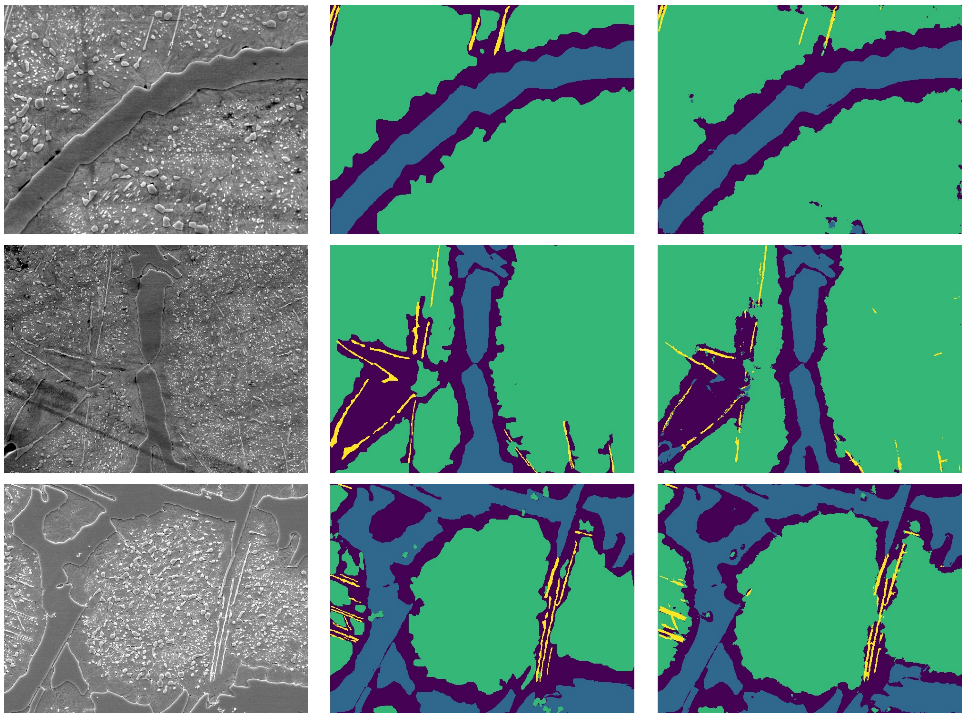
- When is a black box OK?
  - The overall cost of wrong answers is low.
  - The method is better than all alternatives within its domain.
  - The results inspire or guide further inquiry.



E. A. Holm *Science* 364:6435 26-27 (2019)

The overall cost of wrong answers is low:
Autonomous microstructural segmentation using deep learning

- Segmenting complex, multi-component microstructures



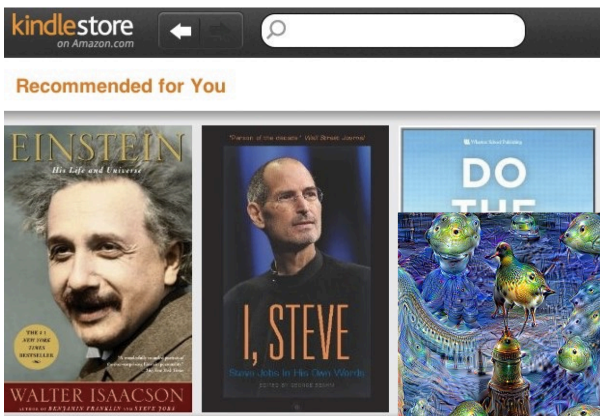
Original Image Grad student Pixel-Net

- Accurate (93 ± 3%)
- Objective
- Repeatable
- Indefatigable
- Permanent
- Exactly as interpretable as a graduate student

DeCost, B., et al. *Microscopy and Microanalysis, 25*(1), 21-29. (2019)

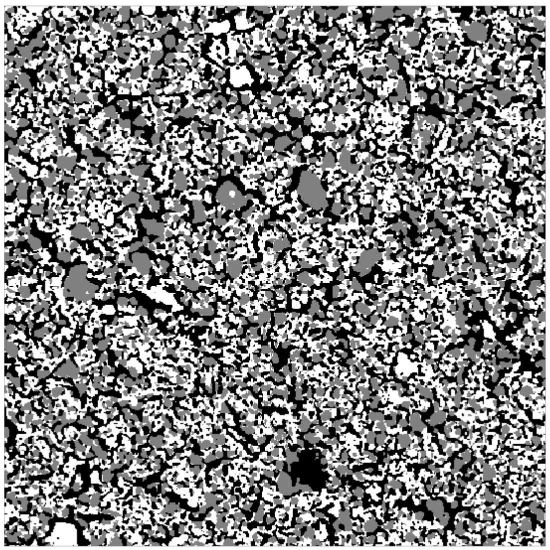
The overall cost of wrong answers is low

- Targeted advertising and recommendations
- **Content organization and analysis**
- Content generation
- Bots

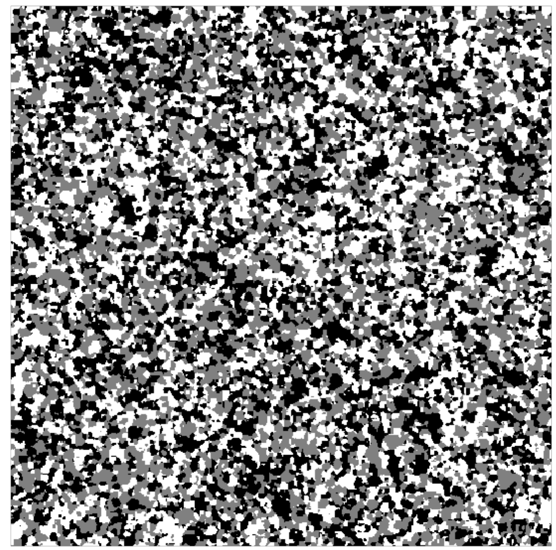


The method is better than all alternatives within its domain: Building better microstructures

"Can we build better synthetic microstructures with ML?"

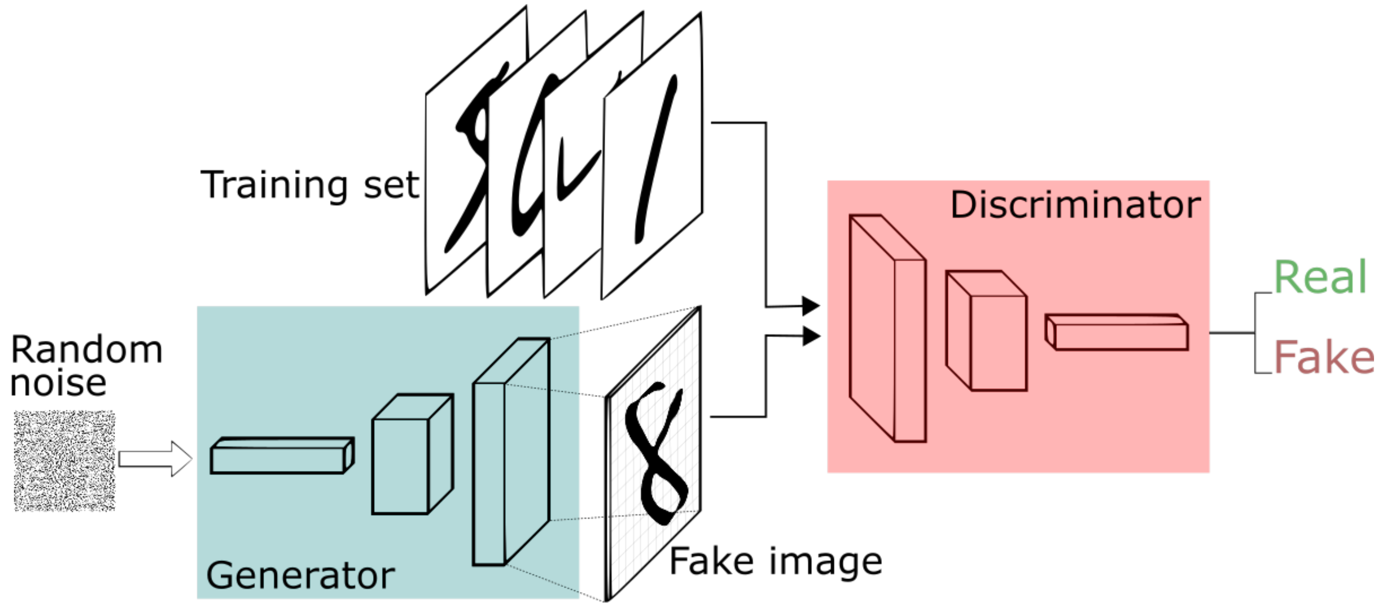


Actual PFIB section of the SOFC cathode



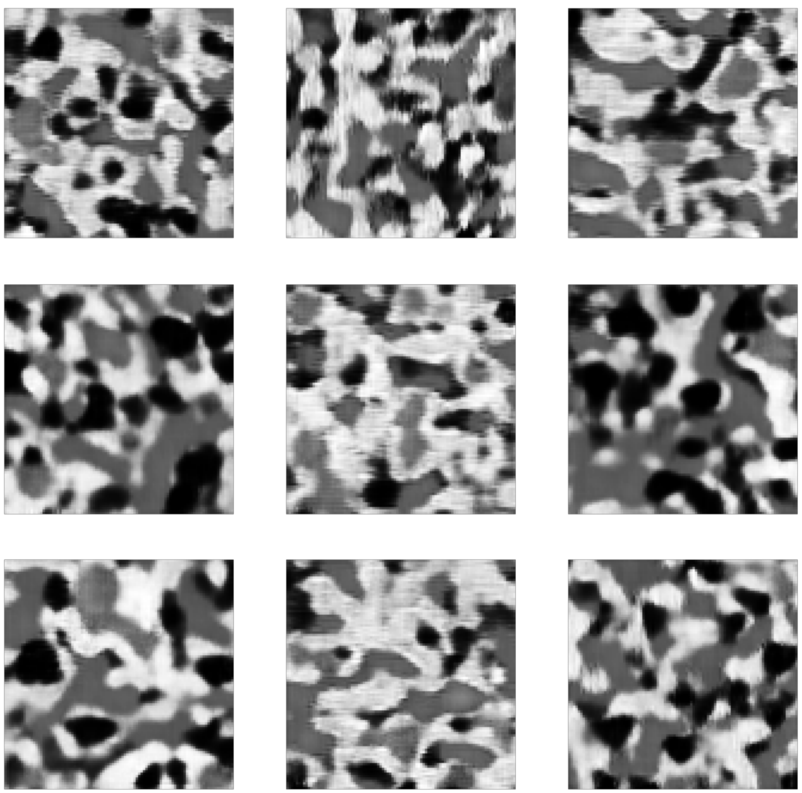
DREAM3d synthetic microstructure

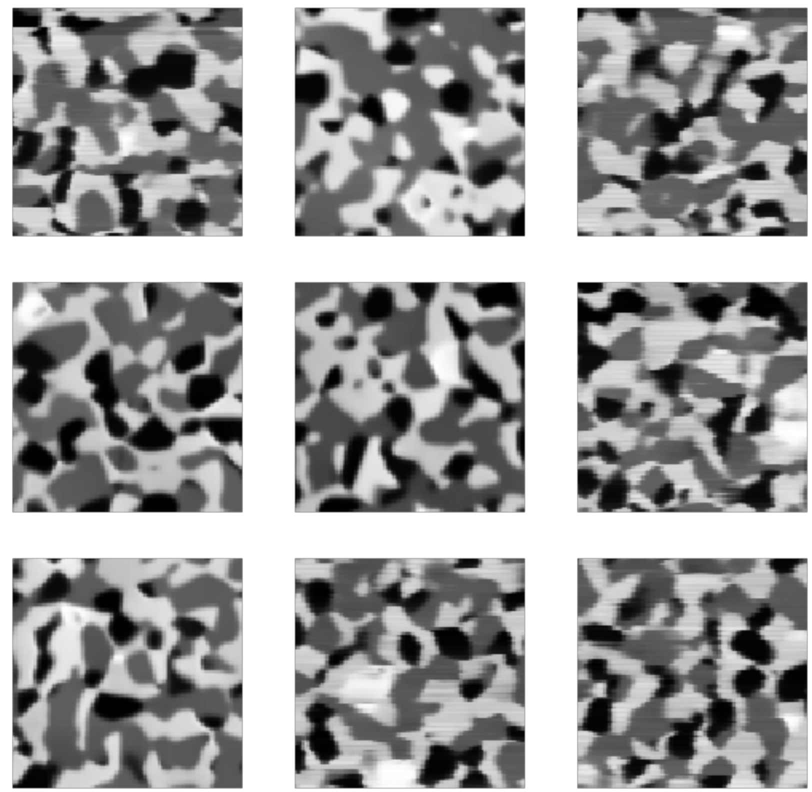
Generative adversarial networks (GANs) can build convincing "fake" images



- Two CNNs compete: One creates images to fool the other.
- Validation is based on visual similarity (at best) or purely subjective (at worst).

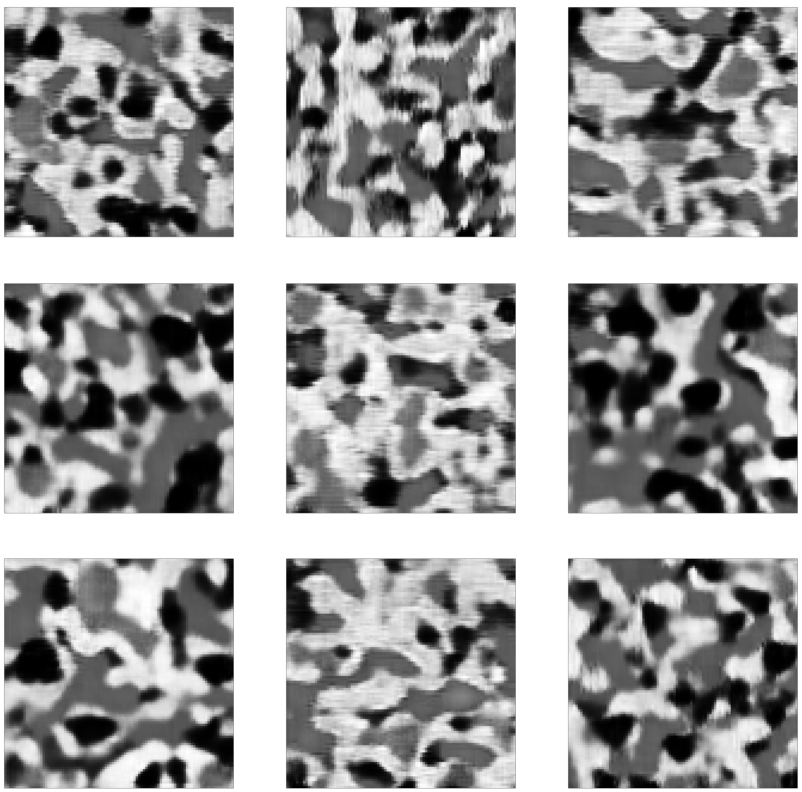
GANs produces convincing results



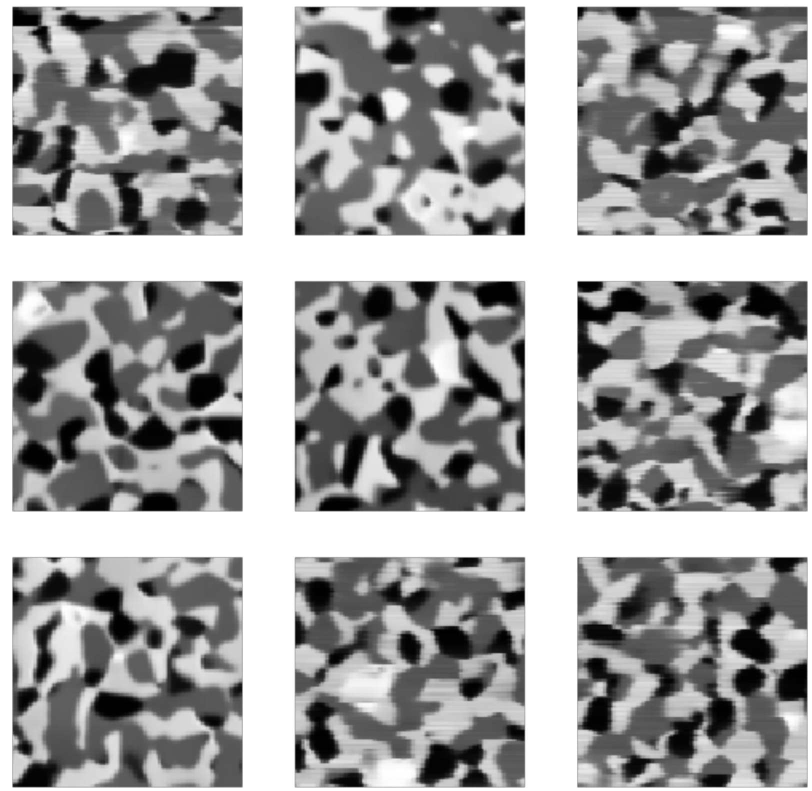


GANs produces convincing results

Fake images

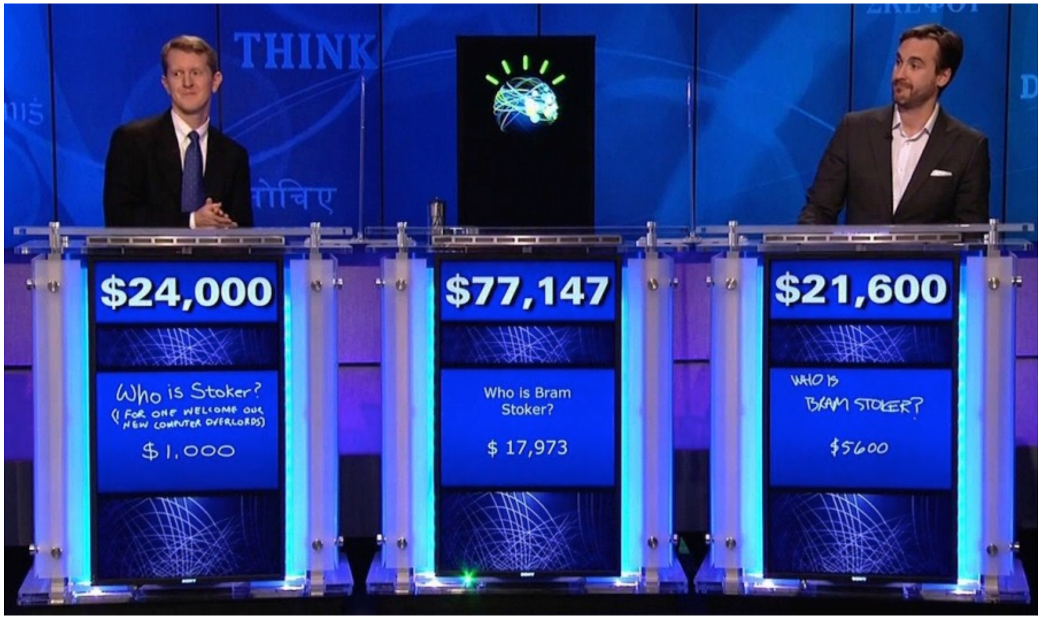


Real images



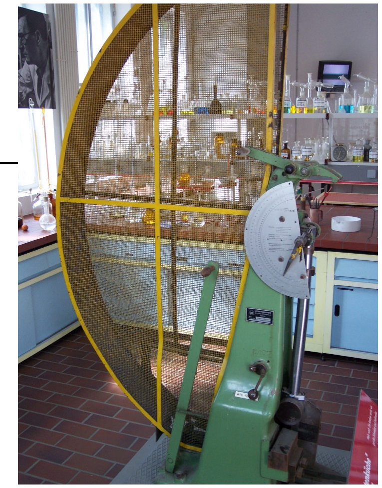
The method is better than all alternatives within its domain

- **Image interpretation – materials, geology, satellite data, …**
- **Autonomy – self-operating equipment, robotics, …**
- **Predictions – actuarial, financial markets, opportunity identification, …**
- **Analysis - quality control, process optimization, data mining, …**
- Security – computer, personal data, access control, …

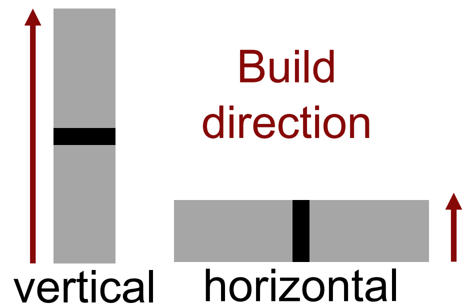


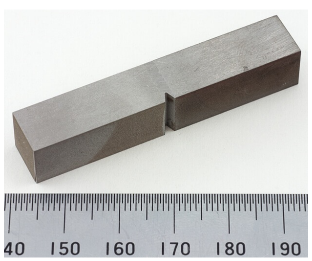
The key is due diligence.

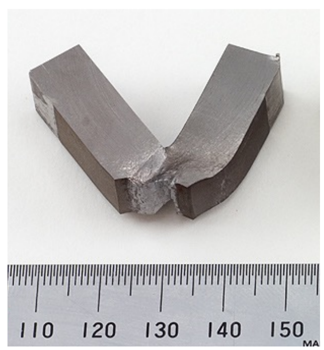
The results inspire or guide further inquiry: Discovering a visual signature for fracture energy

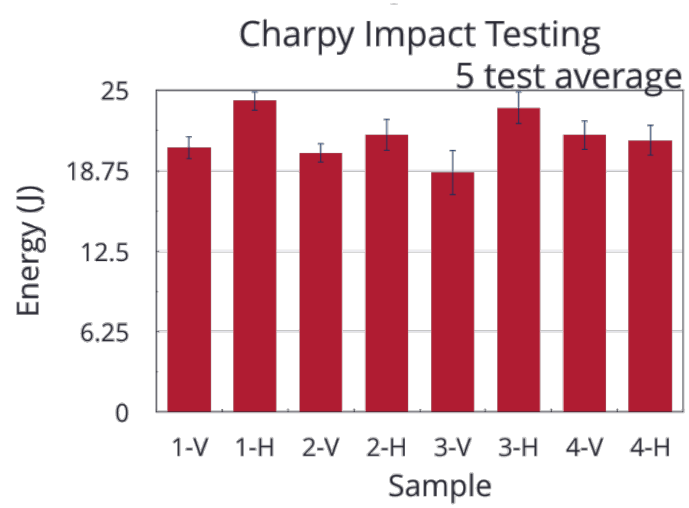


- Inconel 718 Charpy impact specimens built using additive manufacturing.
- Two build orientations, horizontal and vertical.
- Charpy impact energy depends on build orientation.

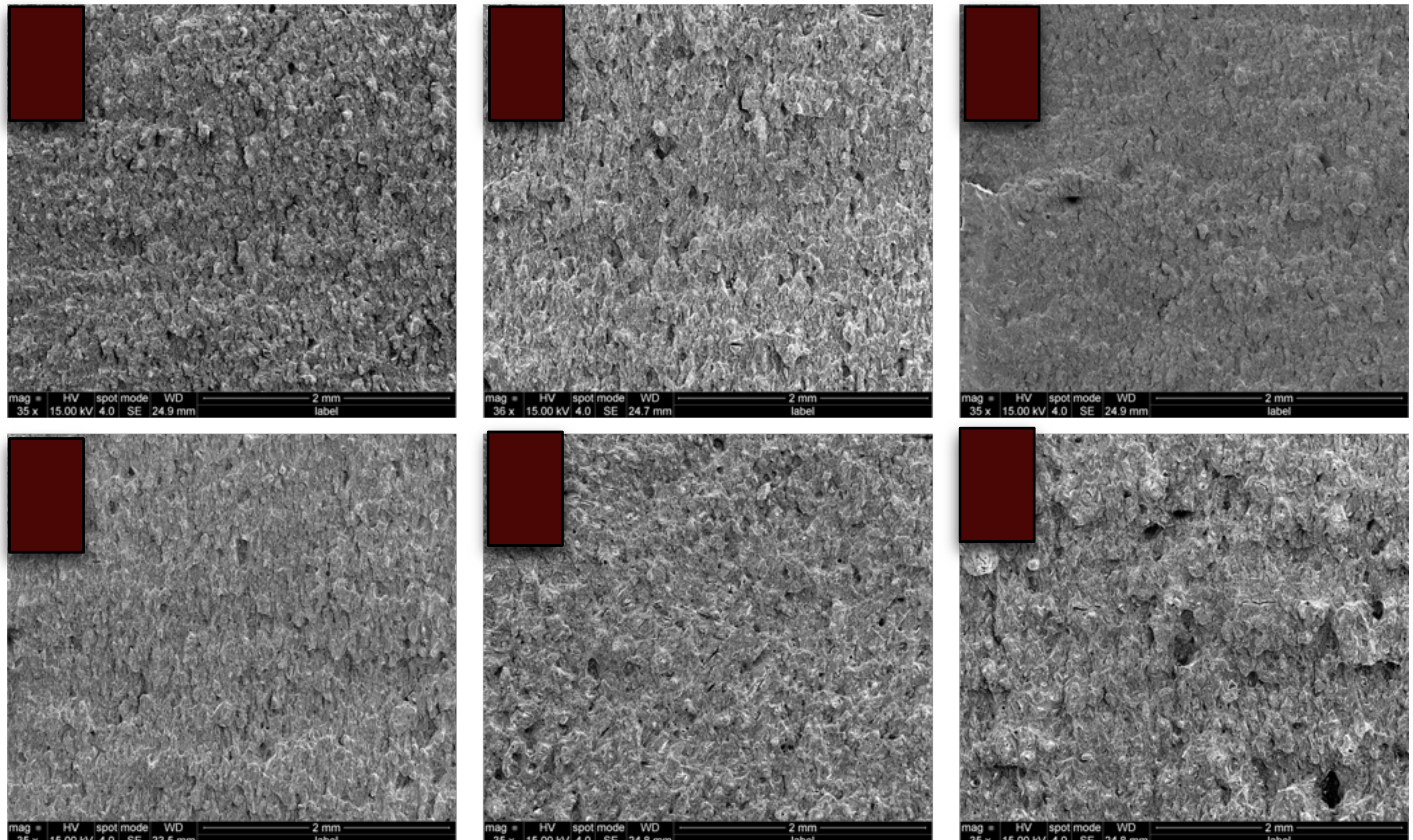






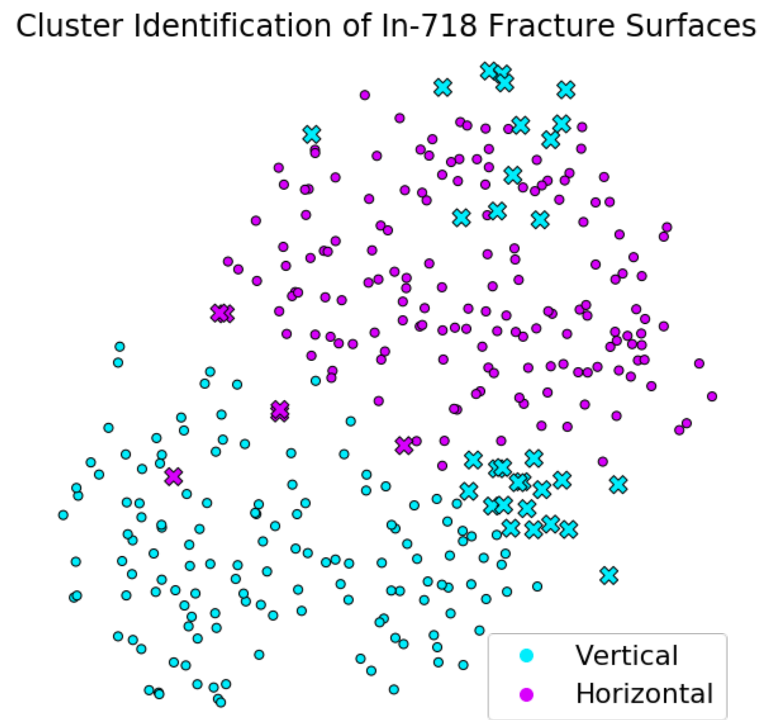


Can you see the difference in the fracture surfaces?



What is the computer learning?

- CNN identified 4096 features, which were reduced using PCA. 50 dimensions were kept (60% of variance), followed by t-SNE and k-means clustering. The computer can identify horizontal and vertical build fractures with **88 ± 3 % accuracy**.
- **What does the computer see that we cannot?**
- **Does the distinguishing visual information provide physical insight?**
- **Has the computer learned fracture mechanics?**



A. R. Kitahara, et al., *IMMI* **7** 148-156 (2018)

The results inspire or guide further inquiry

- Reconsider the spoon 

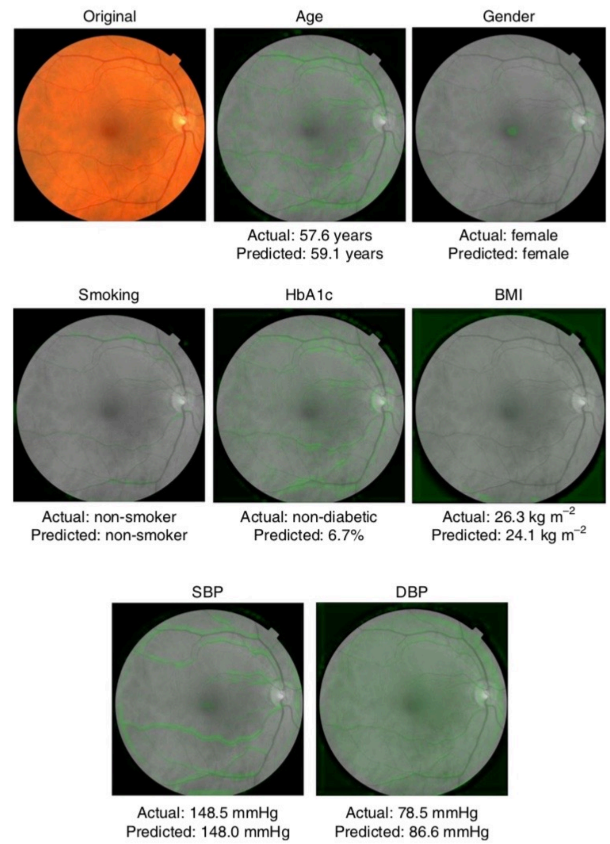
- The top 5 Google Images of chocolate sauce:

- When applied with knowledge, judgment, and responsibility, black box results may **inspire discovery.**

The results inspire or guide further inquiry

- Mining data for unforeseen trends
- Surrogate models for physical simulation
- Make good predictions from incomplete or subresolved data
- Active learning for efficient workflows
- Generate information to augment and extend understanding



R. Poplin, et al., *Nature BME* **2** 158-164 (2018)

To interpret or not to interpret…

- When is a black box OK?
  - The overall cost of wrong answers is low.
  - The method is better than all alternatives within its domain.
  - The results inspire or guide further inquiry.
- **When must we move past the black box?**
  - **The goal is insight: causation, systematization, understanding.**

"All right," said Deep Thought. "The Answer to the Great Question..."
"Yes..!"
"Of Life, the Universe and Everything..." said Deep Thought.
"Yes...!"
"Is..." said Deep Thought, and paused.
"Yes...!"
"Is..."
"Yes...!!!...?"
"Forty-two," said Deep Thought, with infinite majesty and calm.

—Douglas Adams, The Hitchhiker's Guide to the Galaxy

NEU Defect Database Example

To run the CNN example, look for the zipped package, neu_vgg16_example-master-Feb21.zip, which contains a Jupyter notebook called 1.0-ark_tutorial.ipynb and the (datasets) folder of images with NEU-CLS.zip.

From the website for the NEU example: "In the Northeastern University (NEU) surface defect database, six kinds of typical surface defects of the hot-rolled steel strip are collected, i.e., rolled-in scale (RS), patches (Pa), crazing (Cr), pitted surface (PS), inclusion (In) and scratches (Sc). The database includes 1,800 grayscale images: 300 samples each of six different kinds of typical surface defects."

NEU steel defect discussion, examples: https://akbarikevin.medium.com/neu-surface-defect-dataset-with-tensorflow-api-8753c85fe783

http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html

- We would appreciate it if you cite our works when using the database:
  K. Song and Y. Yan, "*A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects*," ***Applied Surface Science***, vol. 285, pp. 858-864, Nov. 2013.(paper)
- Yu He, Kechen Song, Qinggang Meng, Yunhui Yan, "*An End-to-end Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features*," ***IEEE Transactions on Instrumentation and Measurement***, 2020,69(4),1493-1504..(paper)
- Hongwen Dong, Kechen Song, Yu He, Jing Xu, Yunhui Yan, Qinggang Meng, "*PGA-Net: Pyramid Feature Fusion and Global Context Attention Network for Automated Surface Defect Detection*," ***IEEE Transactions on Industrial Informatics***, 2020.(paper)

Questions?