Input and output shapes

(for an image classification example)

We're going to be building CNNs to do this part!

224

224

[[0.31, 0.62, 0.44…],
[0.92, 0.03, 0.27…],
[0.25, 0.78, 0.07…],
…,

Inputs

Machine Learning Algorithm

Outputs

[0.00, 0.97, 0.03]

(prediction probabilities)

(gets represented as a tensor)

[batch_size, width, height, colour_channels]

Shape = [None, 224, 224, 3]
or
Shape = [32, 224, 224, 3]

(32 is a very common batch size)

Shape = [3]

These will vary depending on the problem you're working on.

Improving a model

(from a data perspective)

Method to improve a model (reduce overfitting)	**What does it do?**
More data	Gives a model more of a chance to learn patterns between samples (e.g. if a model is performing poorly on images of pizza, show it more images of pizza).
Data augmentation	Increase the diversity of your training dataset without collecting more data (e.g. take your photos of pizza and randomly rotate them 30°). Increased diversity forces a model to learn more generalisation patterns.
Better data	Not all data samples are created equally. Removing poor samples from or adding better samples to your dataset can improve your model's performance.
Use transfer learning	Take a model's pre-learned patterns from one problem and tweak them to suit your own problem. For example, take a model trained on pictures of cars to recognise pictures of trucks.

The input of image

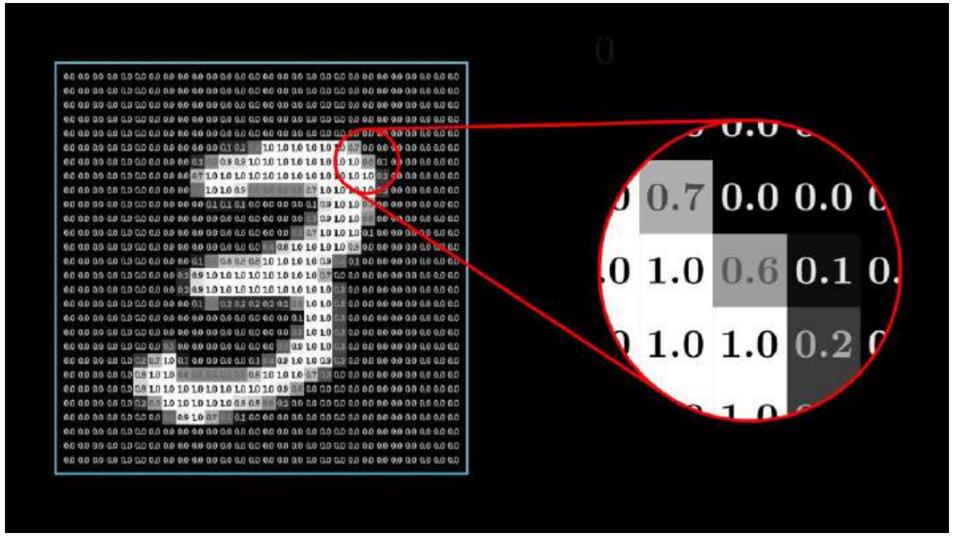
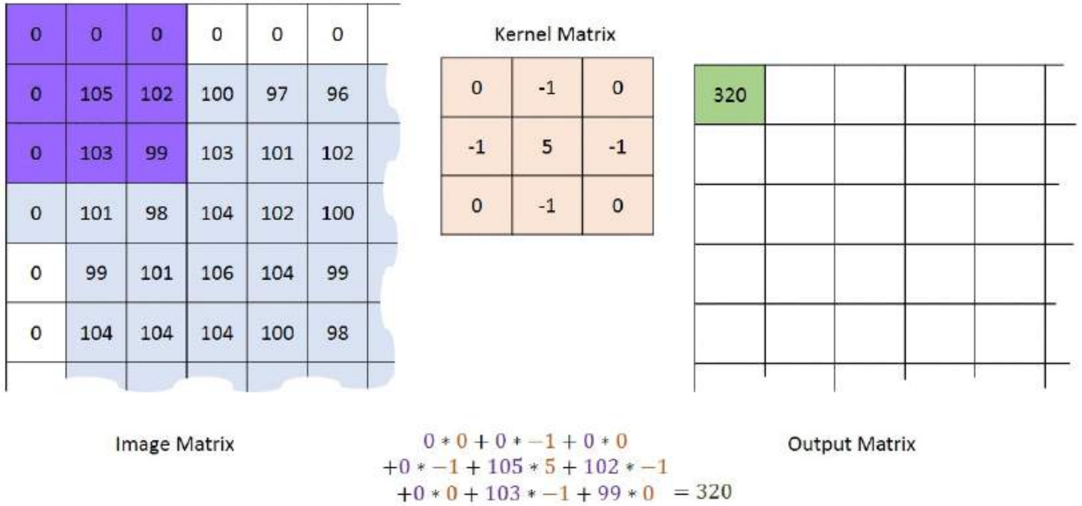
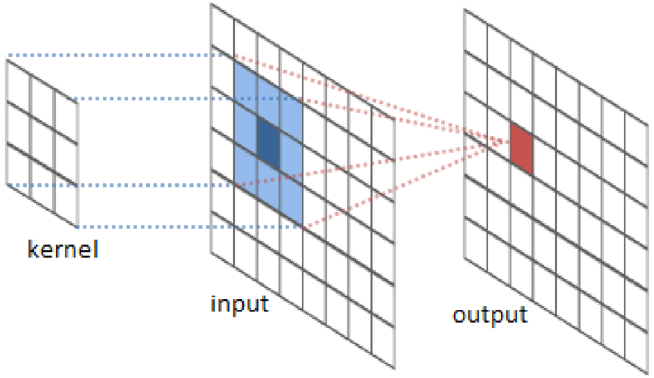


Image with 28 x 28 pixel = consist 784 Neuron

Pixel acted as neuron with an activation value, sort of analogous to how neurons in the brain can be active or inactive

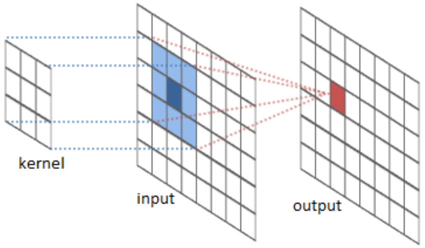
Extracted with filter of kernel matrix





Convolution Operation

Learning the unique features

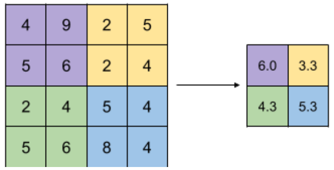


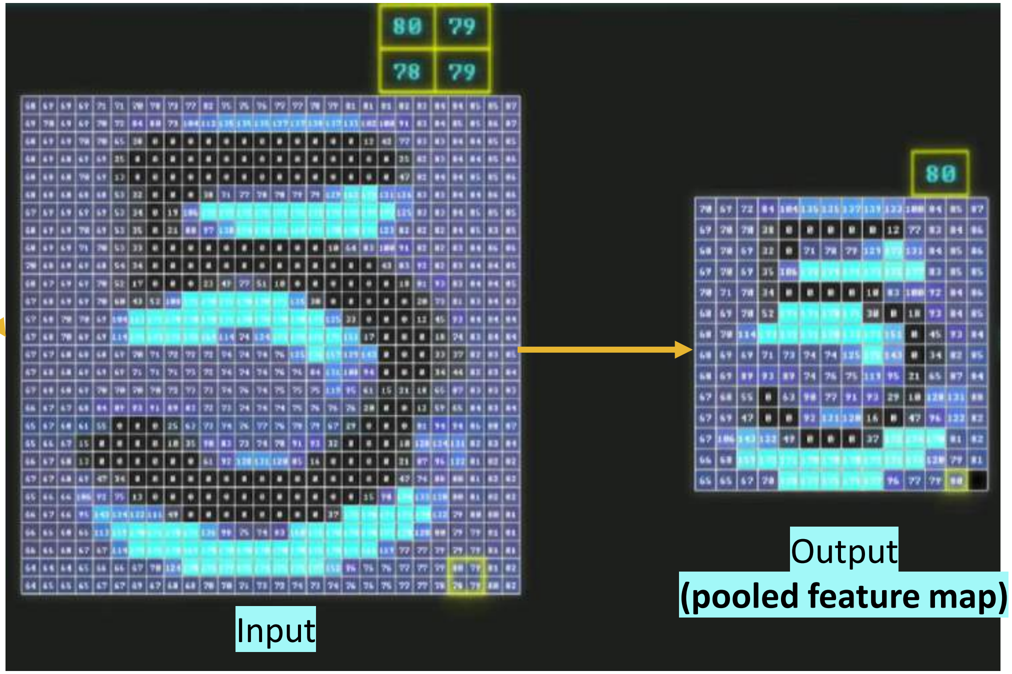
Input

Output
(feature maps)

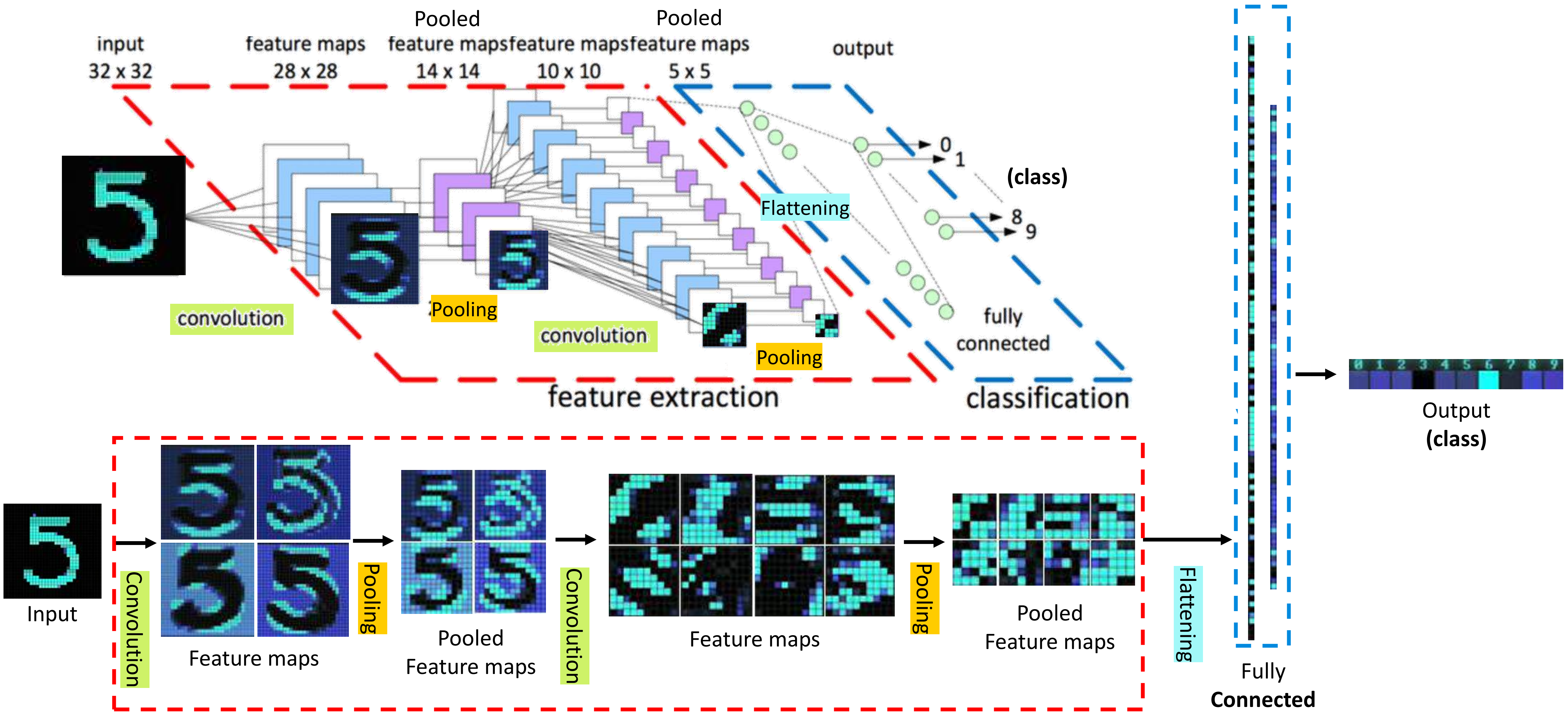
Pooling operation

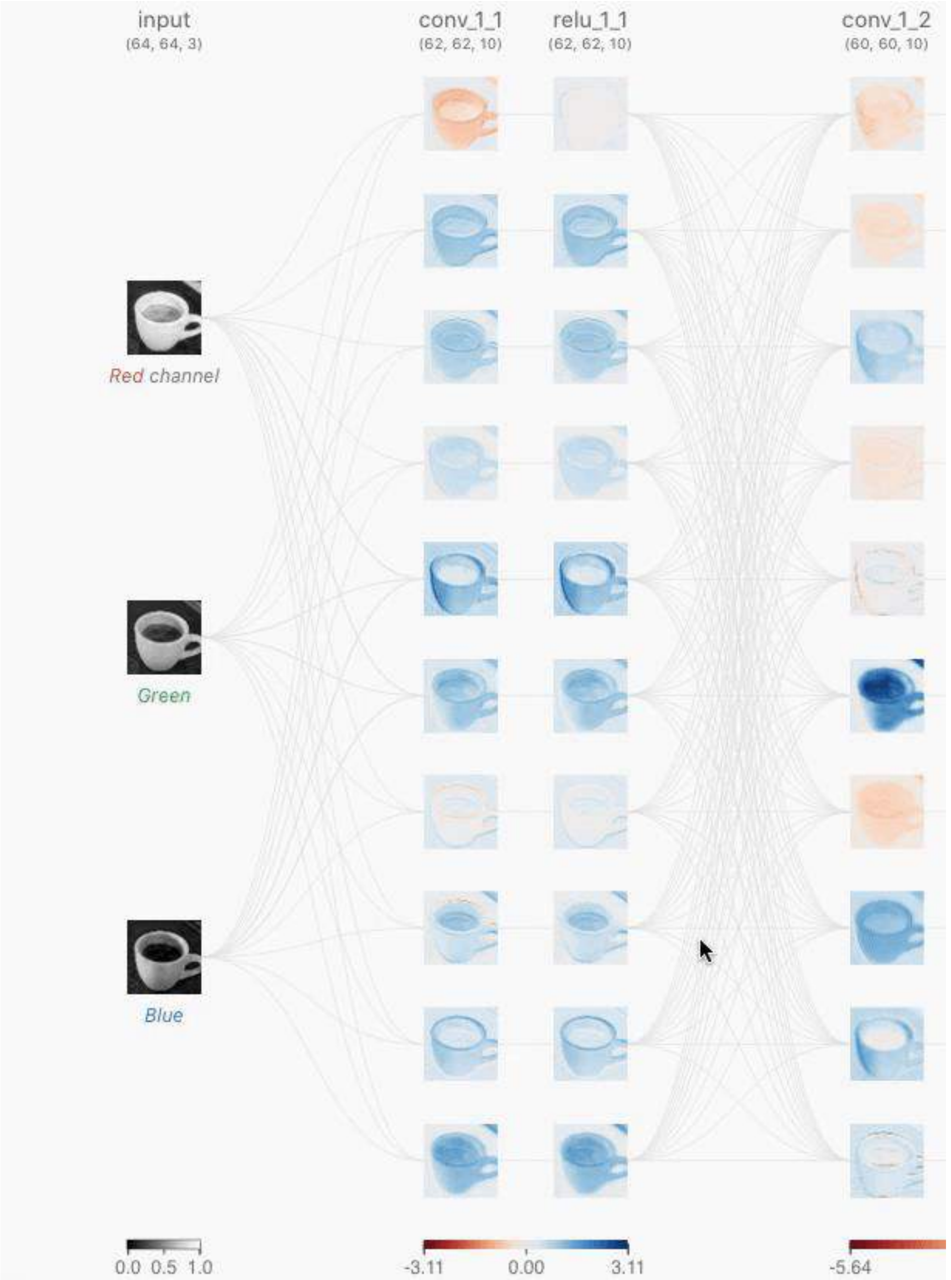
Simplify the data of learned features





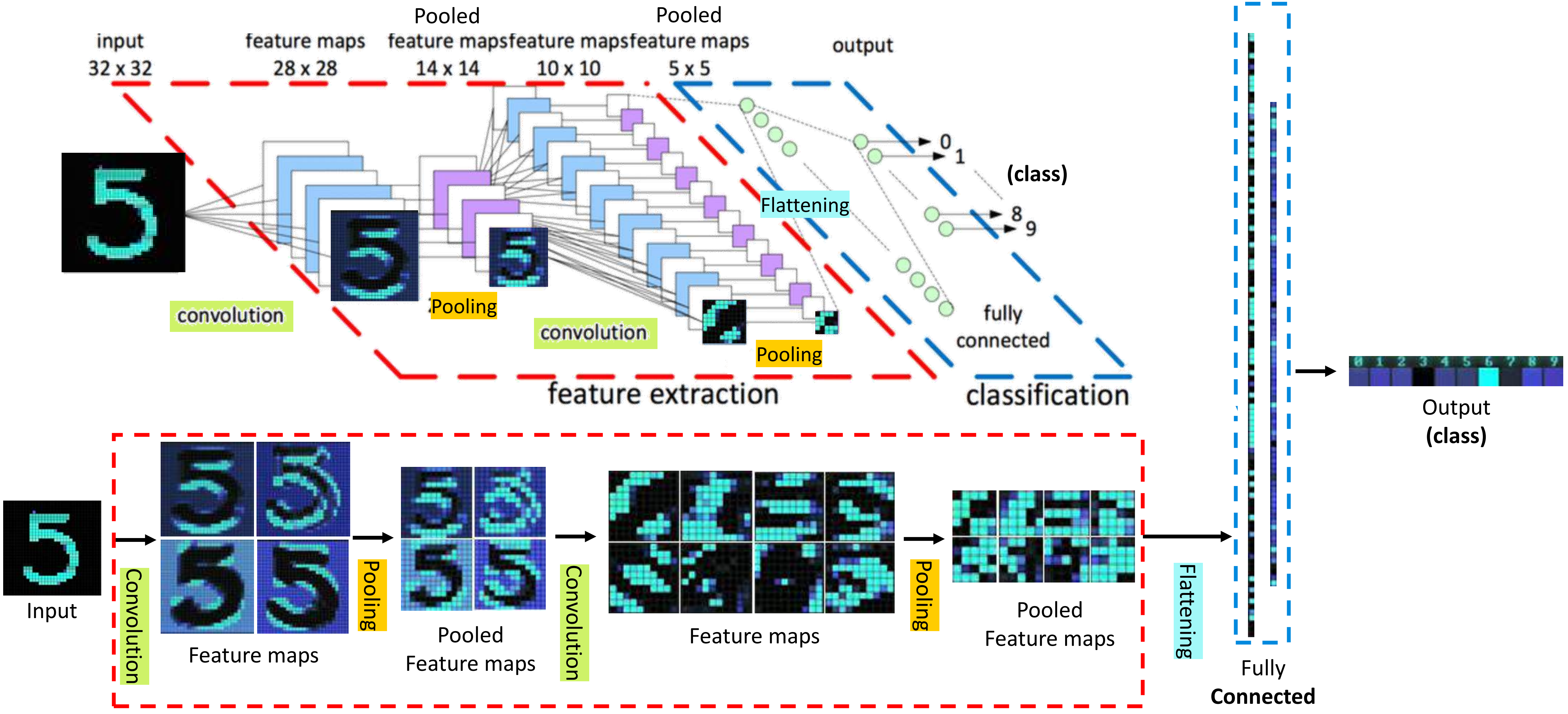
CNN Pipeline



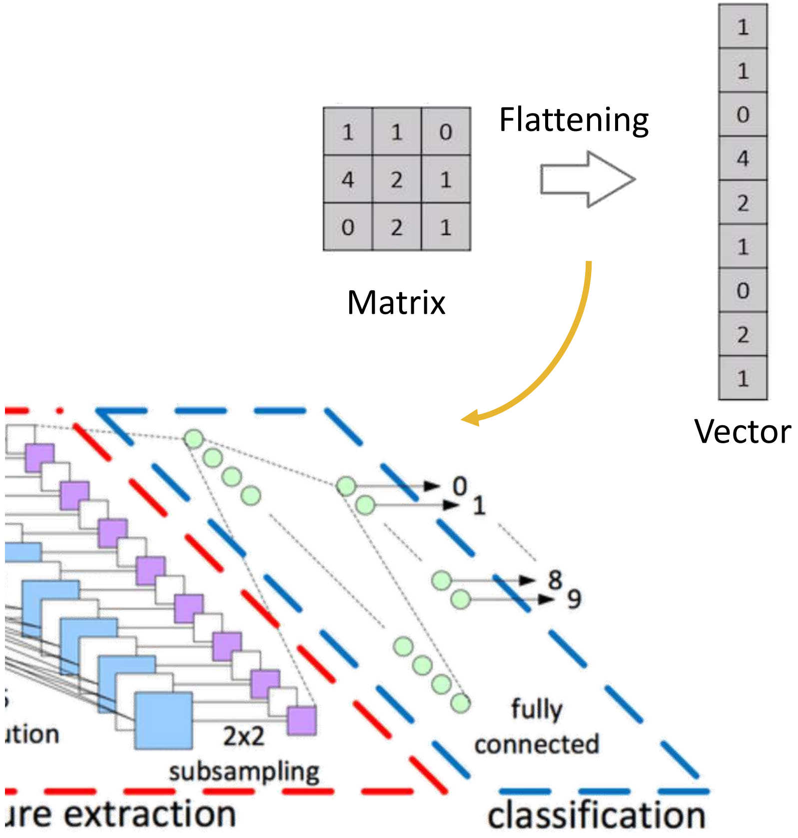


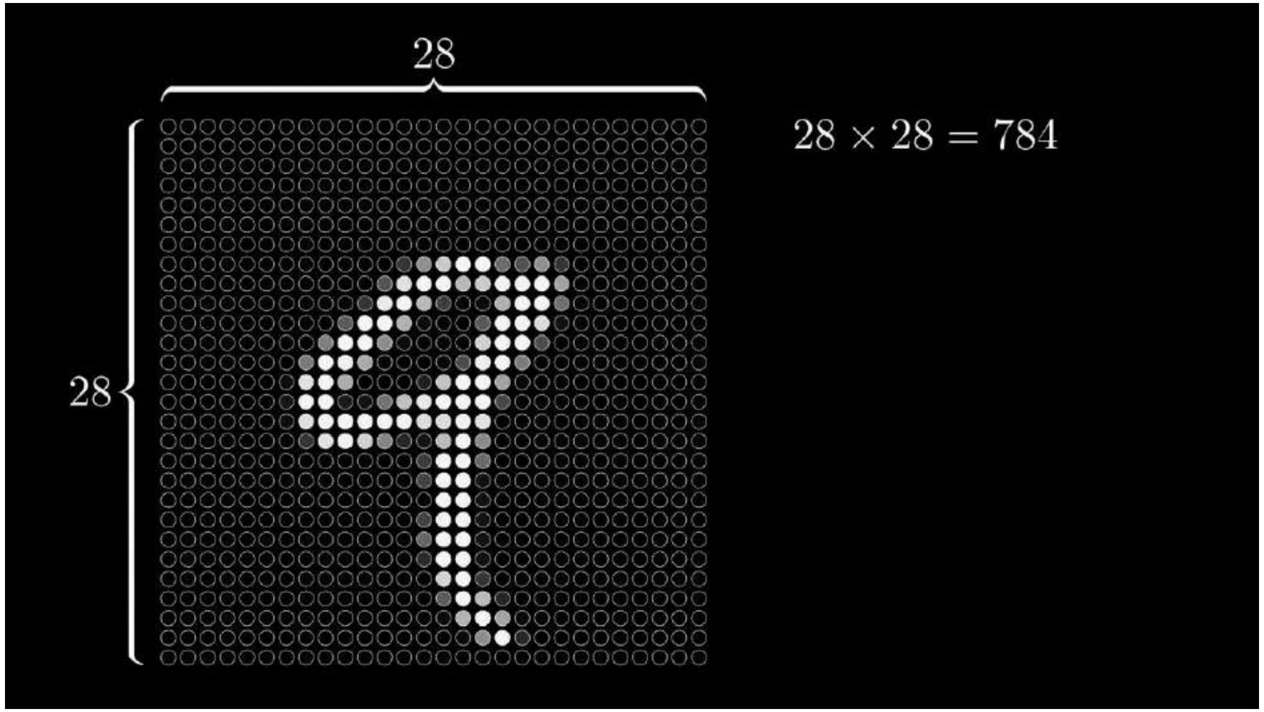
https://poloclub.github.io/cnn-explainer/

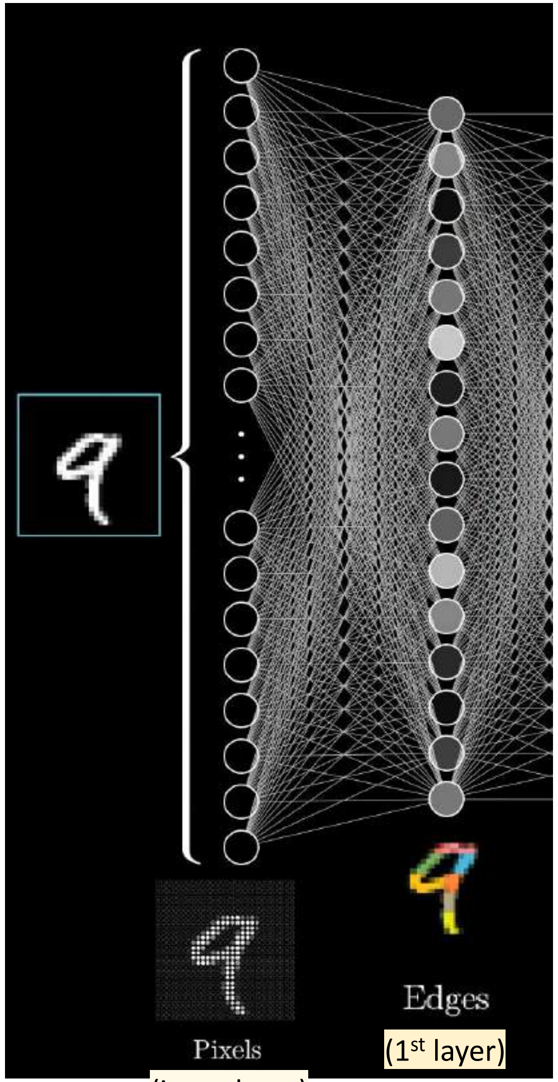
CNN Pipeline



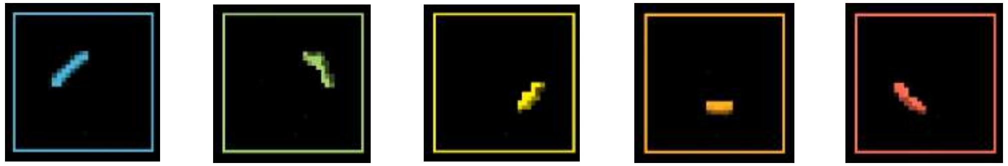
Fully Connected layer





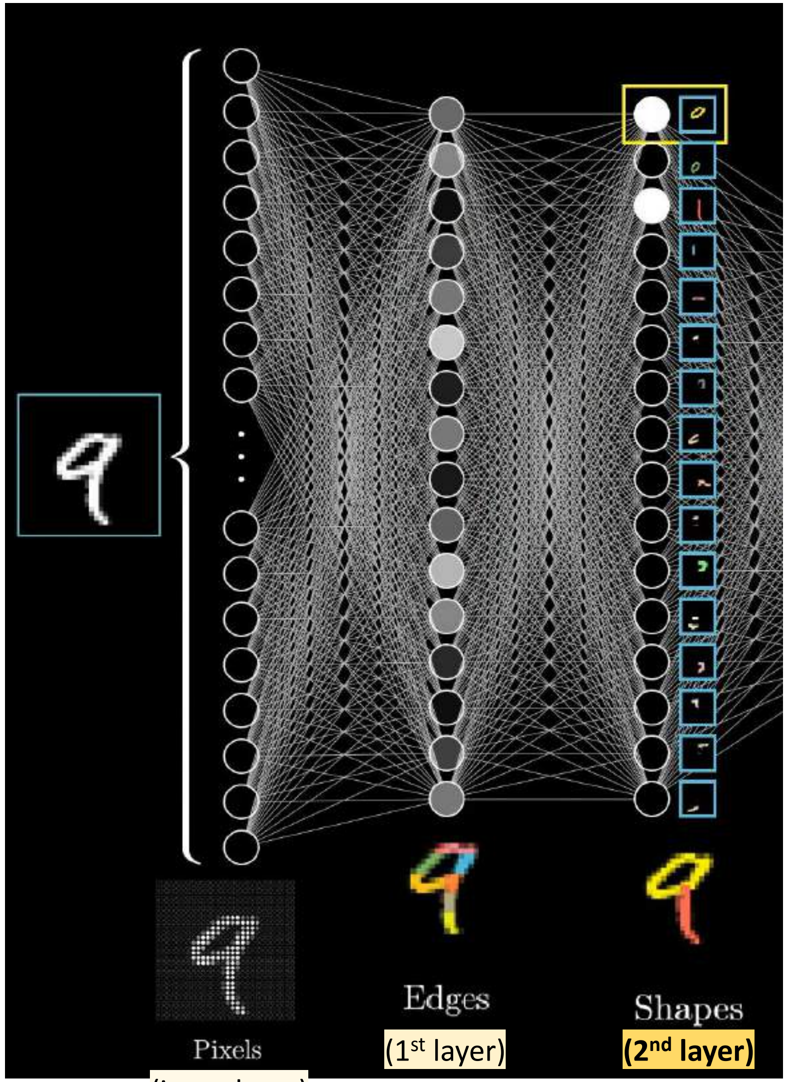


(input layer)

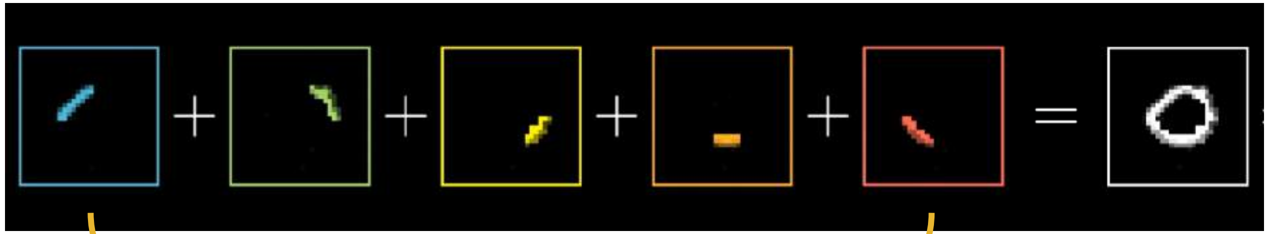


Edges (1st layer)

Edges (1st layer)

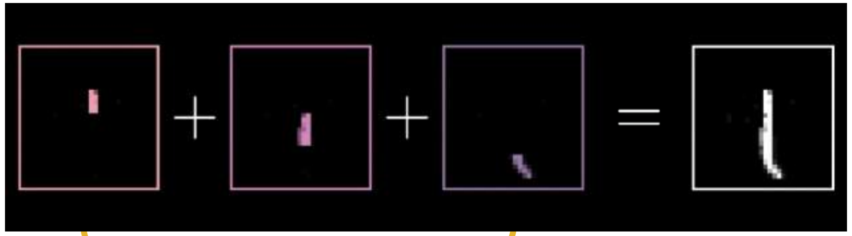


(input layer)



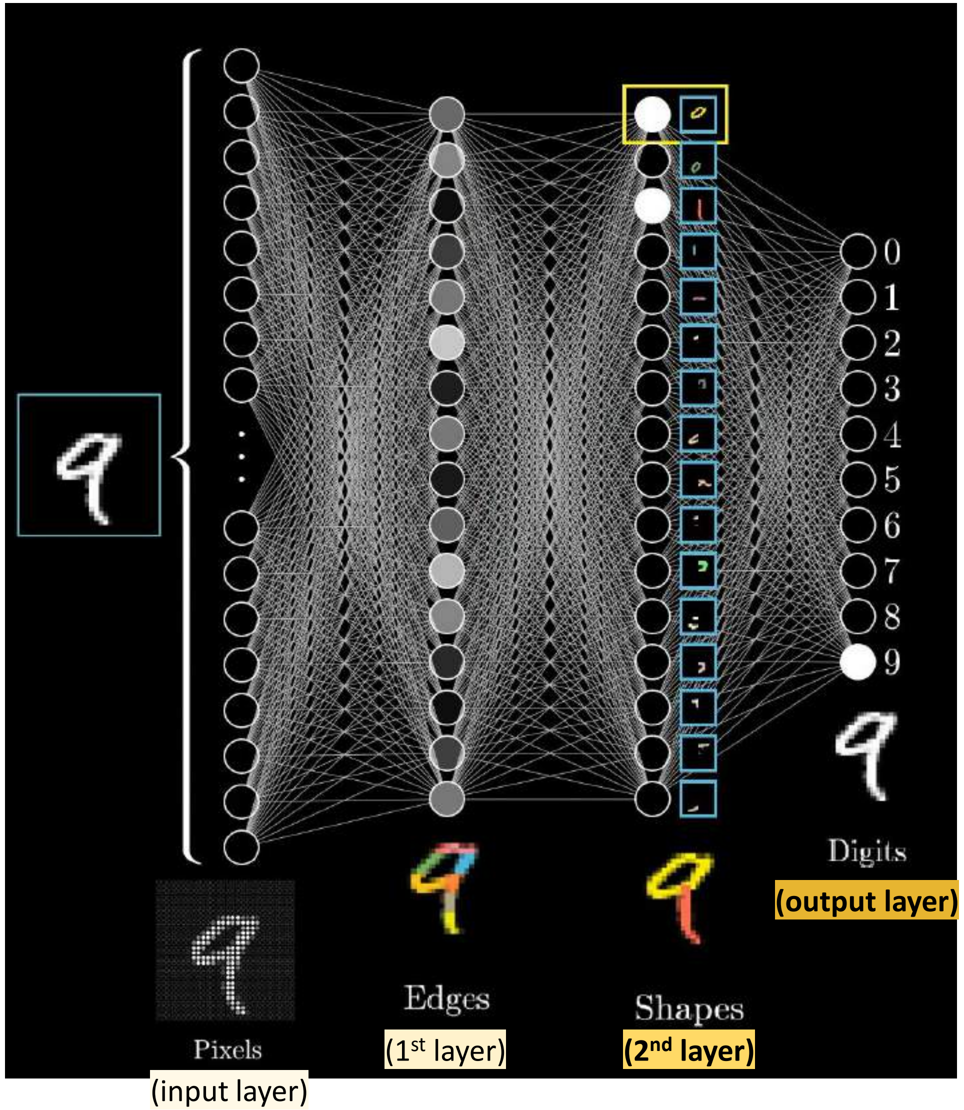
Edges (1st layer)

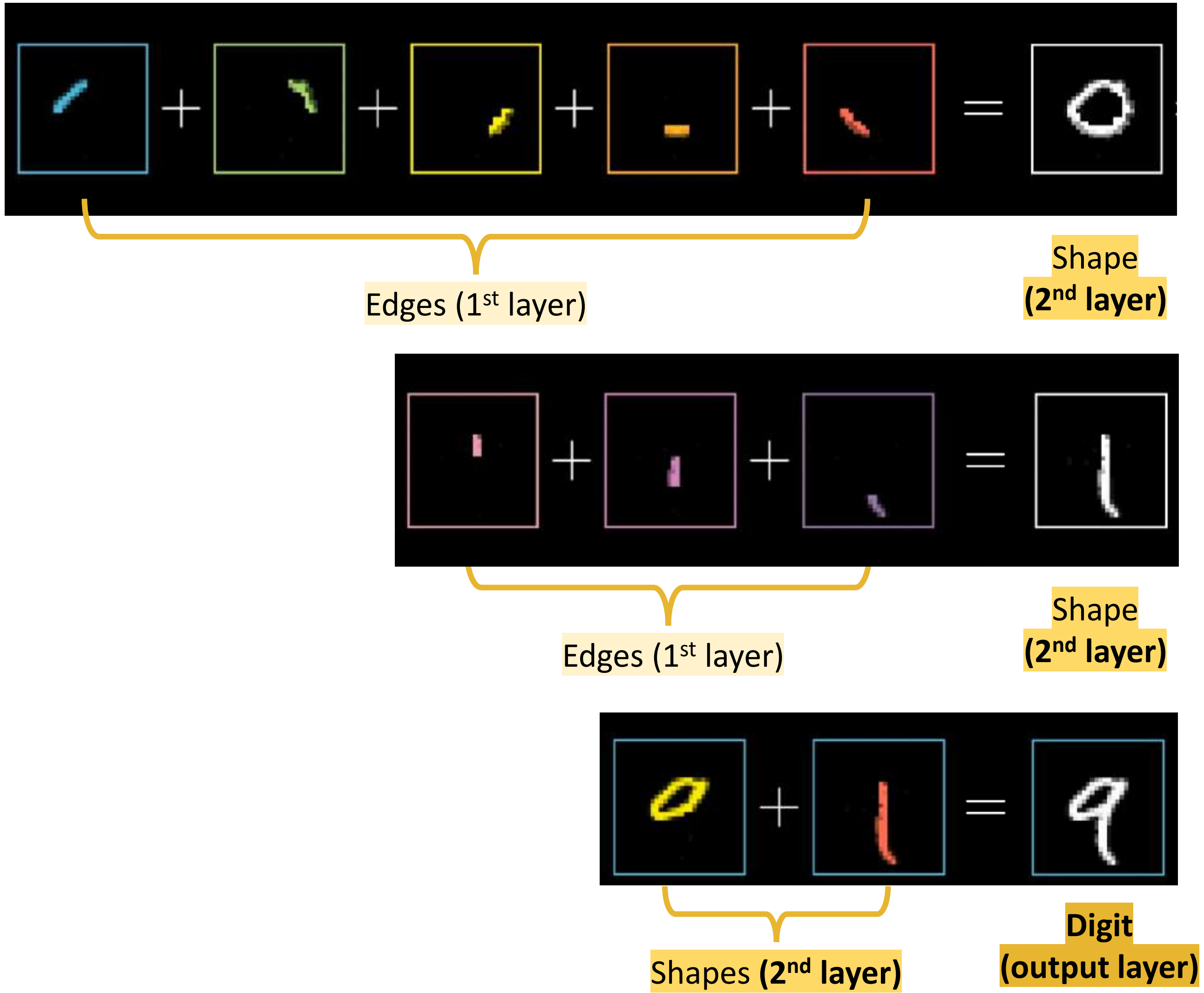
Shape
(2nd layer)

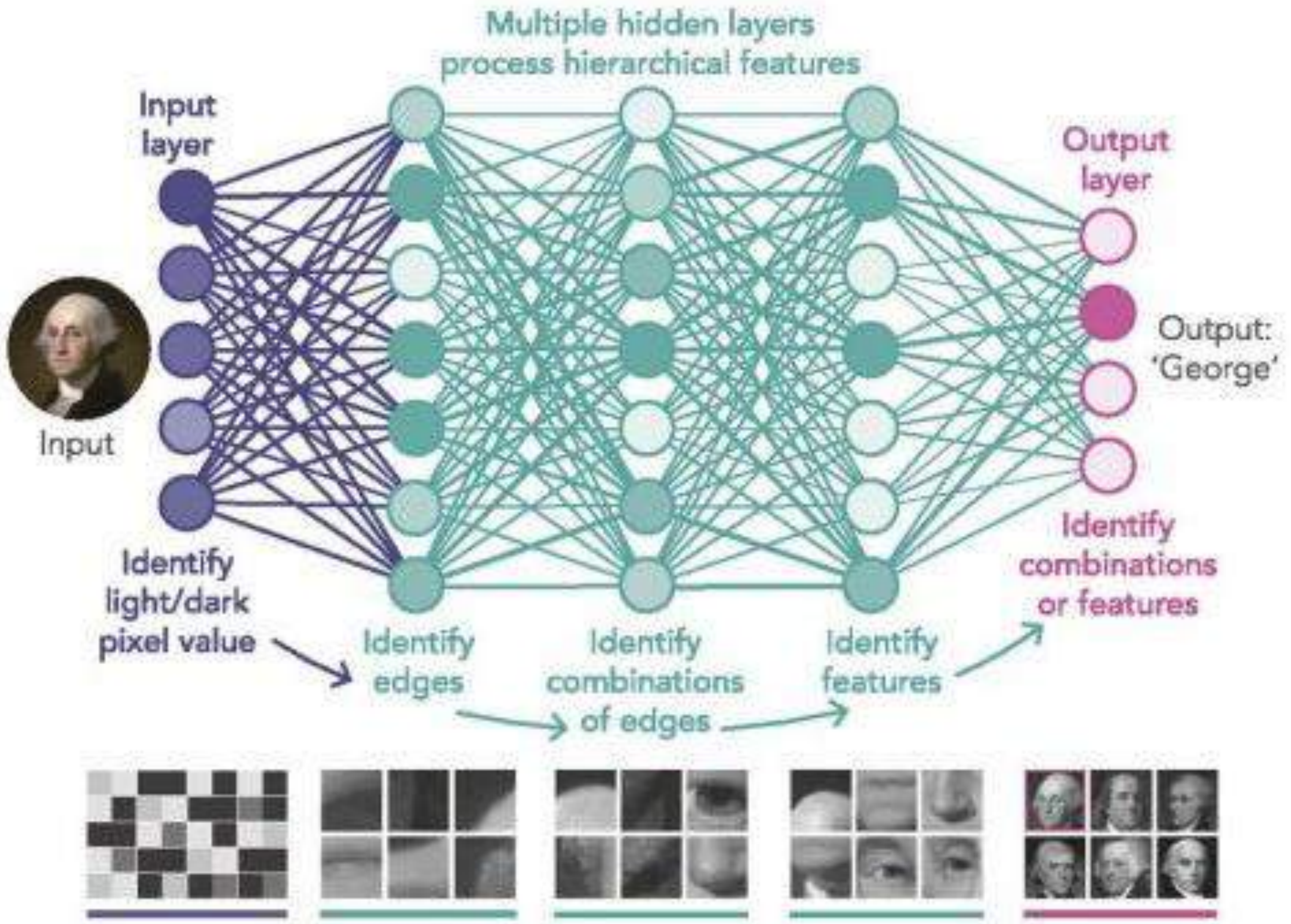


Edges (1st layer)

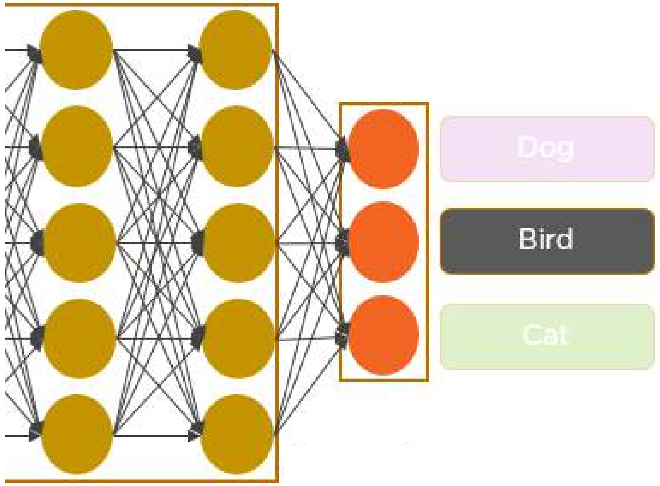
Shape
(2nd layer)

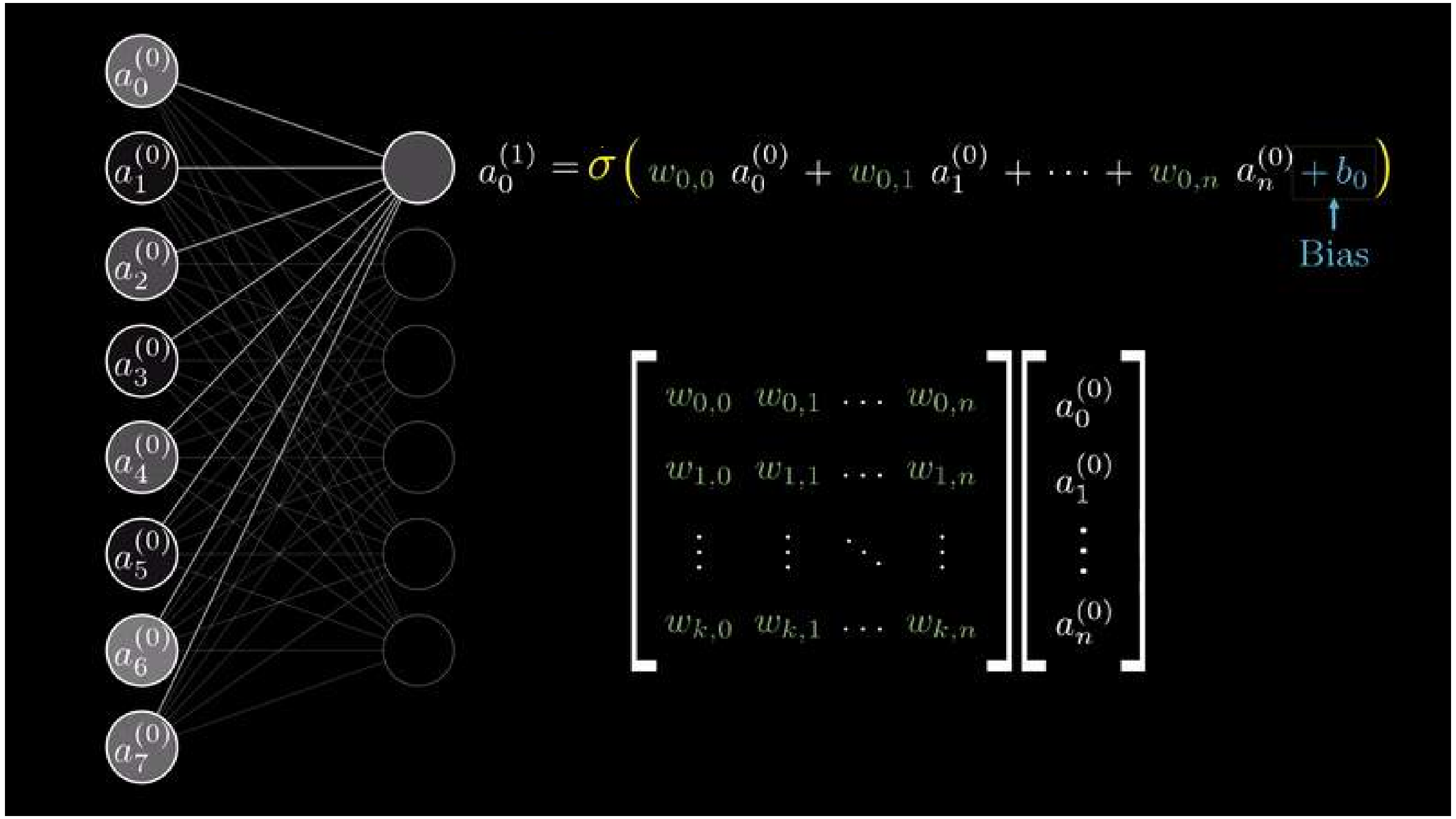

0
1
2
3
4
5
6
7
8
9
Digits
(output layer)
Pixels
(input layer)
Edges
(1st layer)
Shapes
(2nd layer)


Edges (1st layer)
Shape
(2nd layer)
Edges (1st layer)
Shape
(2nd layer)
Shapes (2nd layer)
Digit
(output layer)

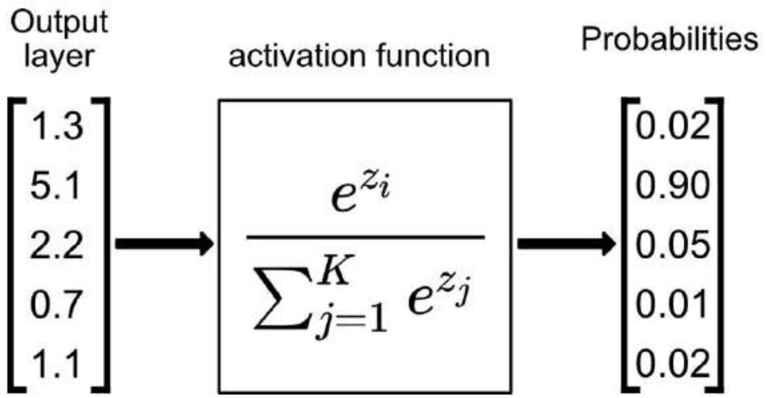

Multiple hidden layers
process hierarchical features
Input
layer
Output
layer
Output:
'George'
Input
Identify
light/dark
pixel value
Identify
edges
Identify
combinations
of edges
Identify
features
Identify
combinations
or features

Output of fully connected layer (model class)





Output of fully connected layer (model class)



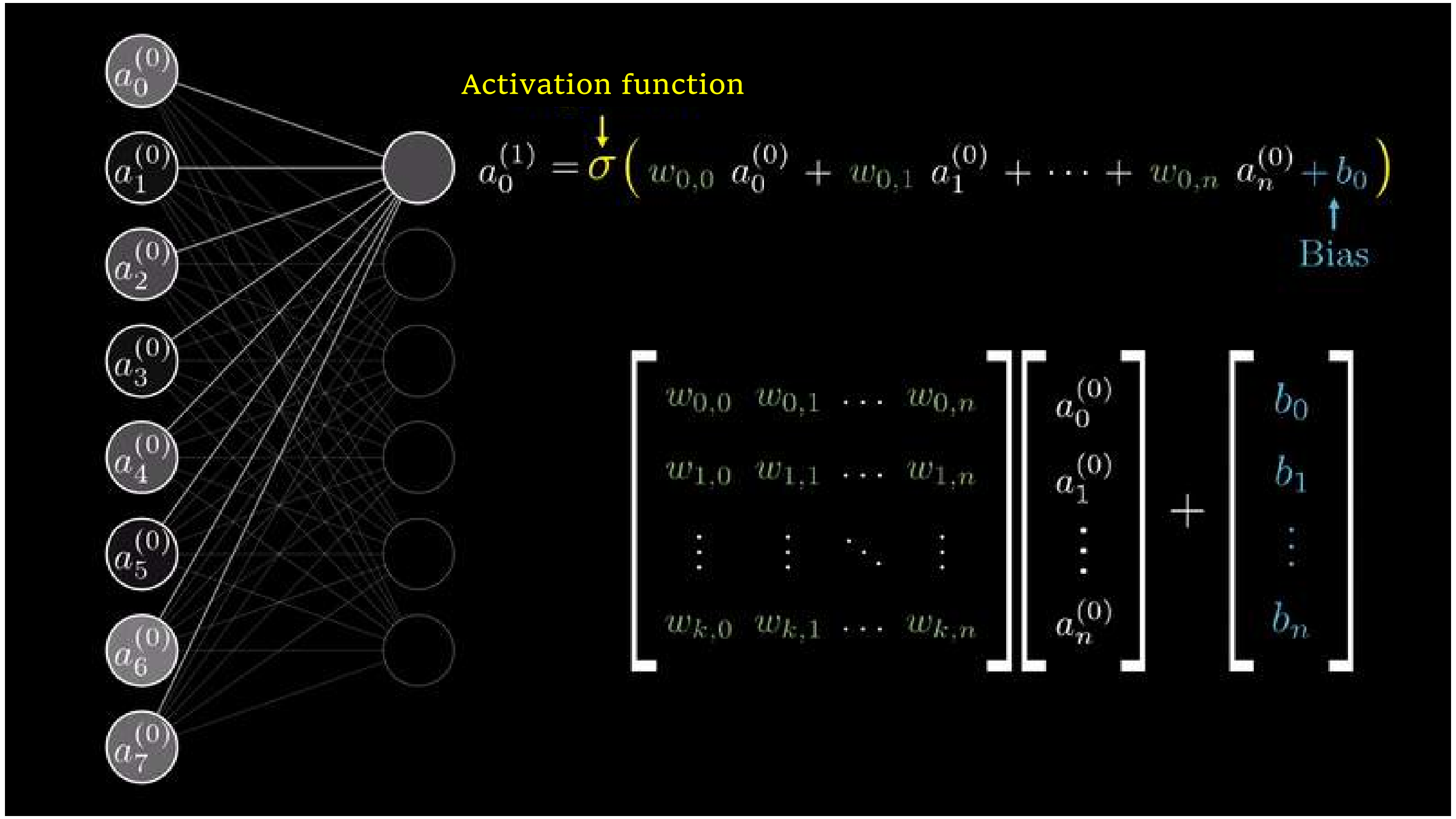
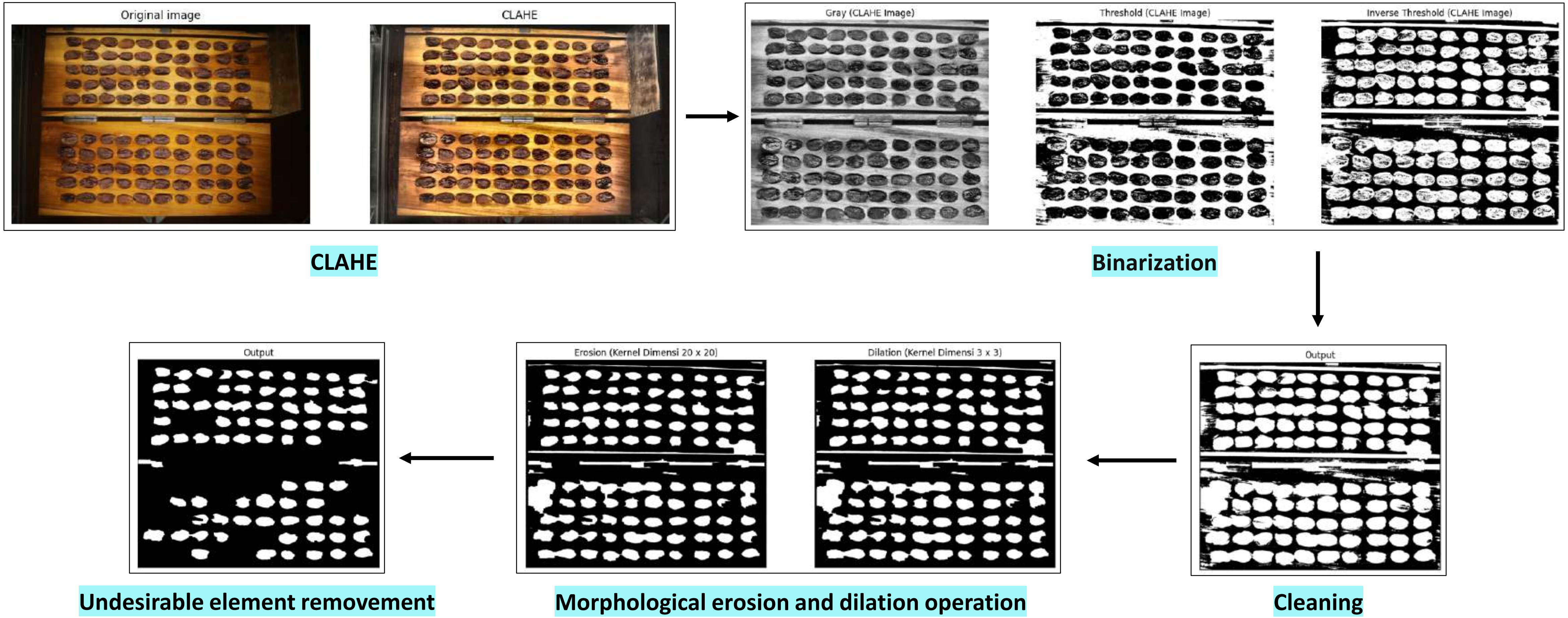


Image pre-processing

Gambar hasil jepretan

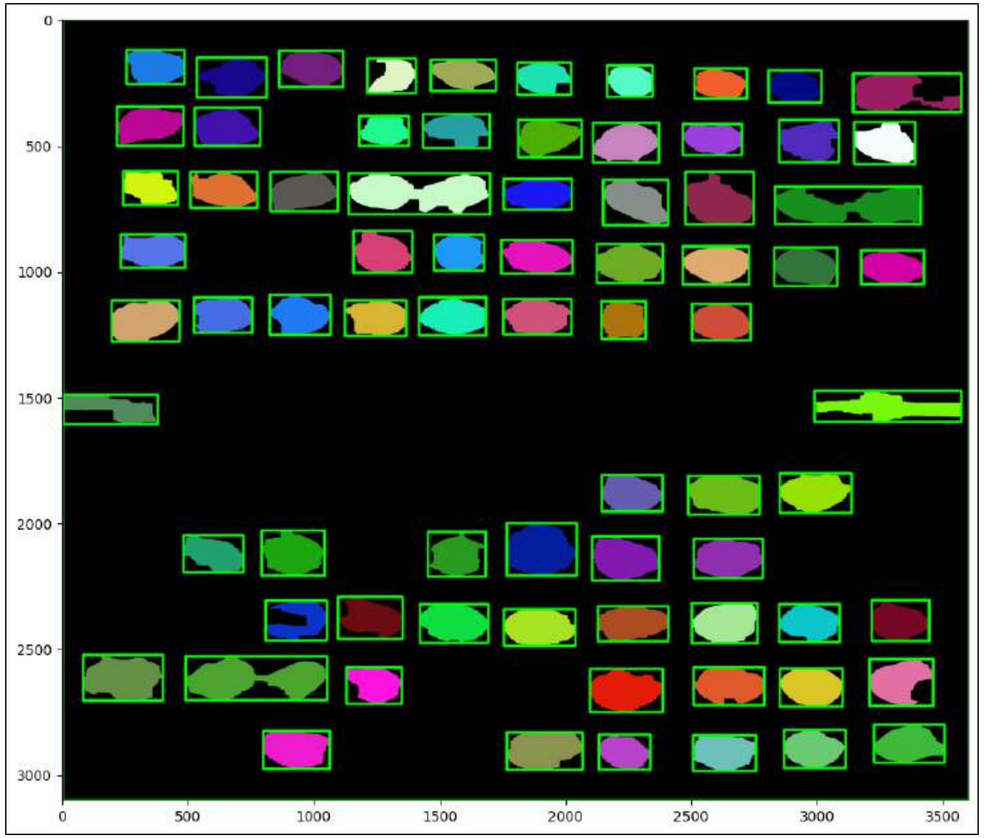
Gambar yang diinginkan model

Image pre-processing



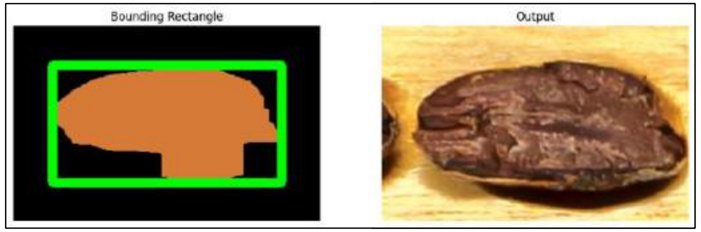
Data pre-processing

Image cropping



Bounding Rectangle

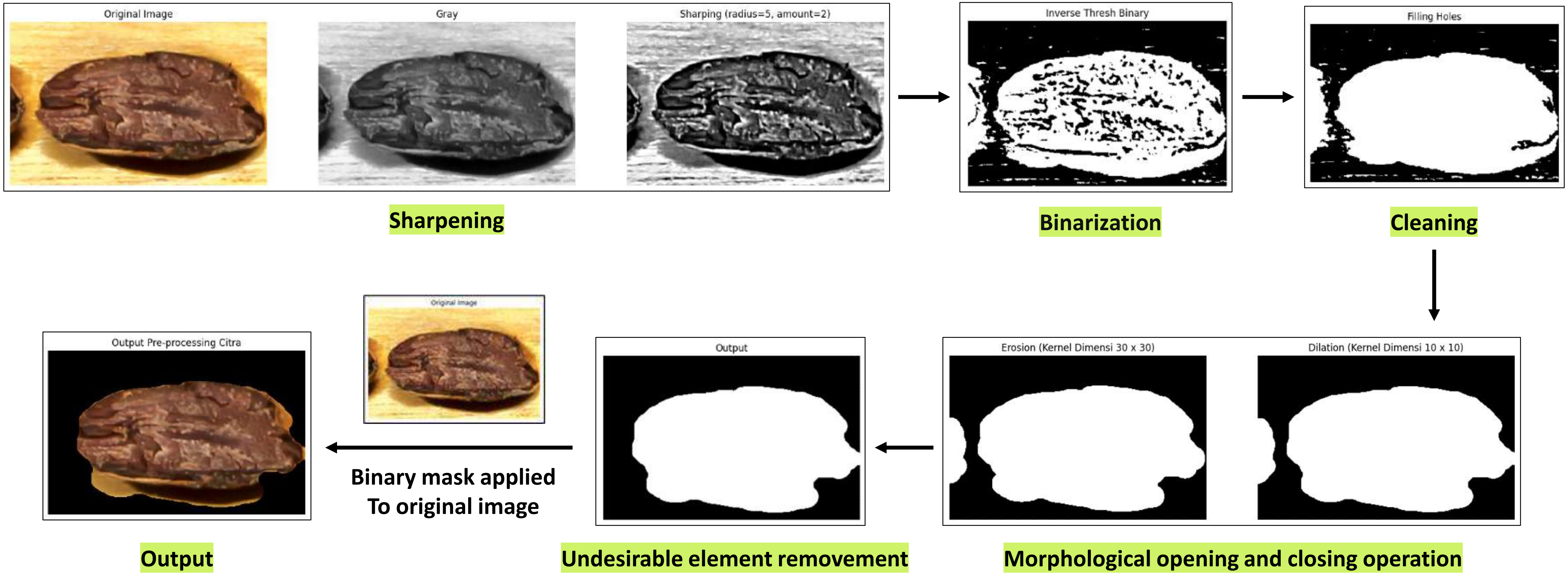
The mask
Applied to original image



Output

Data pre-processing

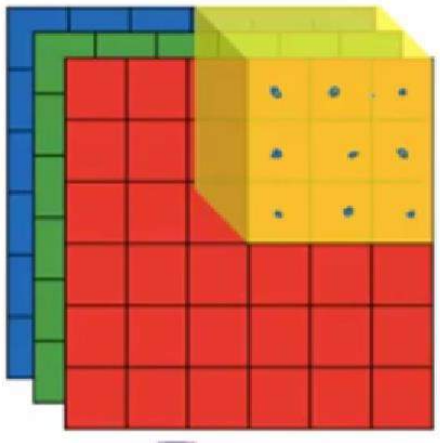
Sample segmentation

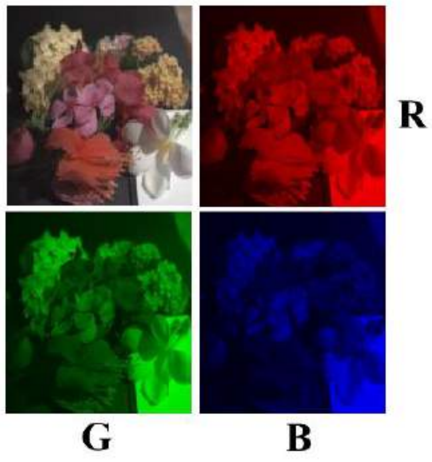


RGB image (gambar berwarna)

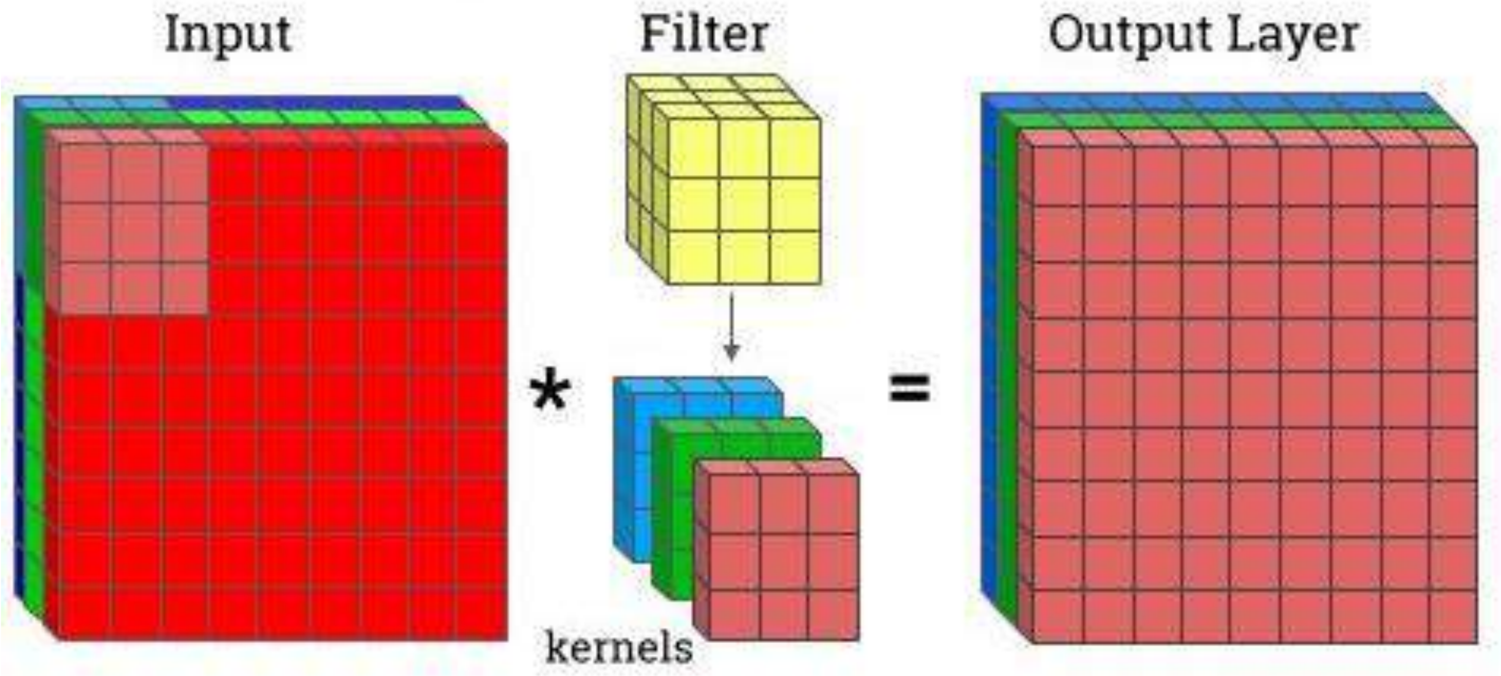
3 channel (dimensi)

- Red channel
- Green channel
- Blue channel



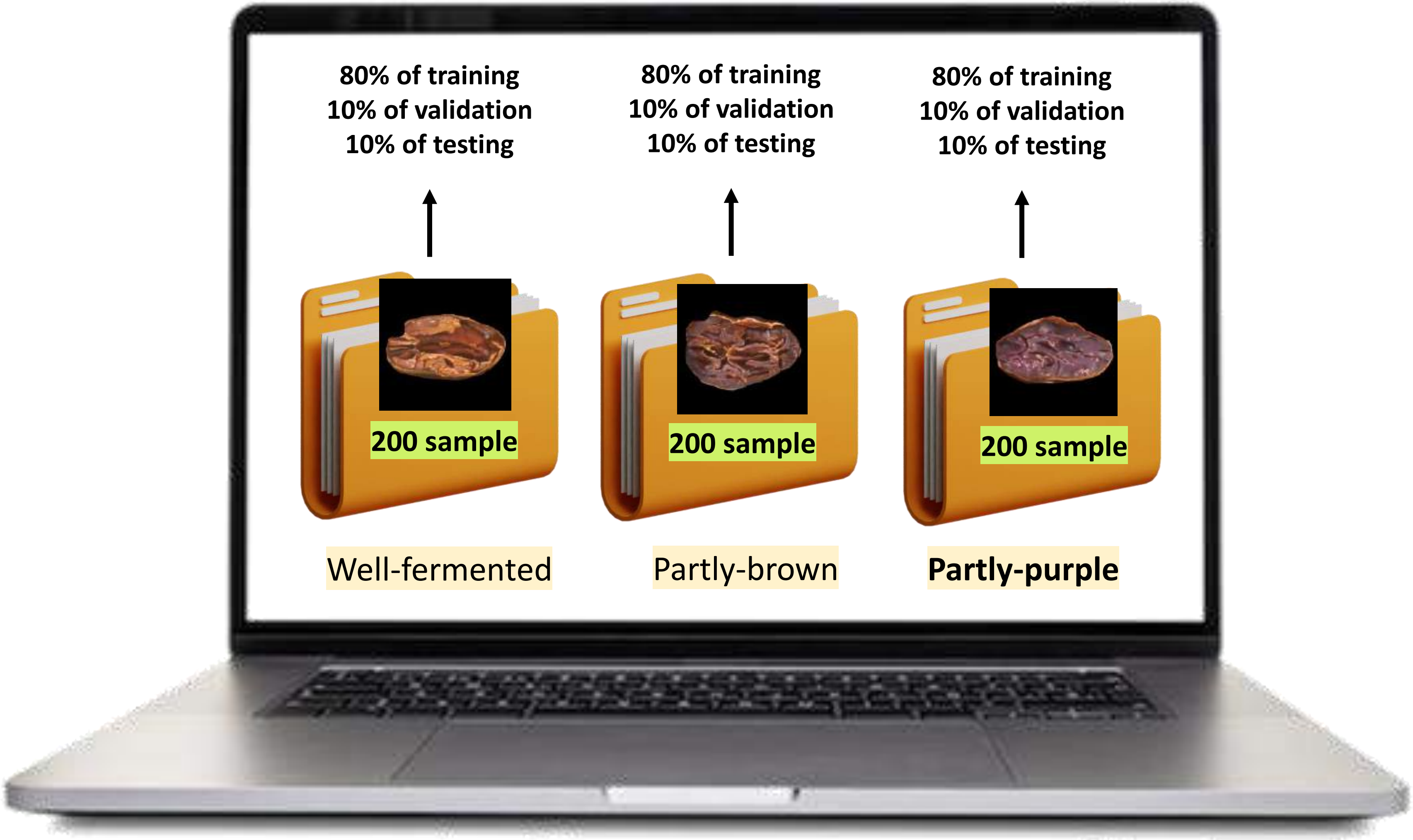


n dimensi

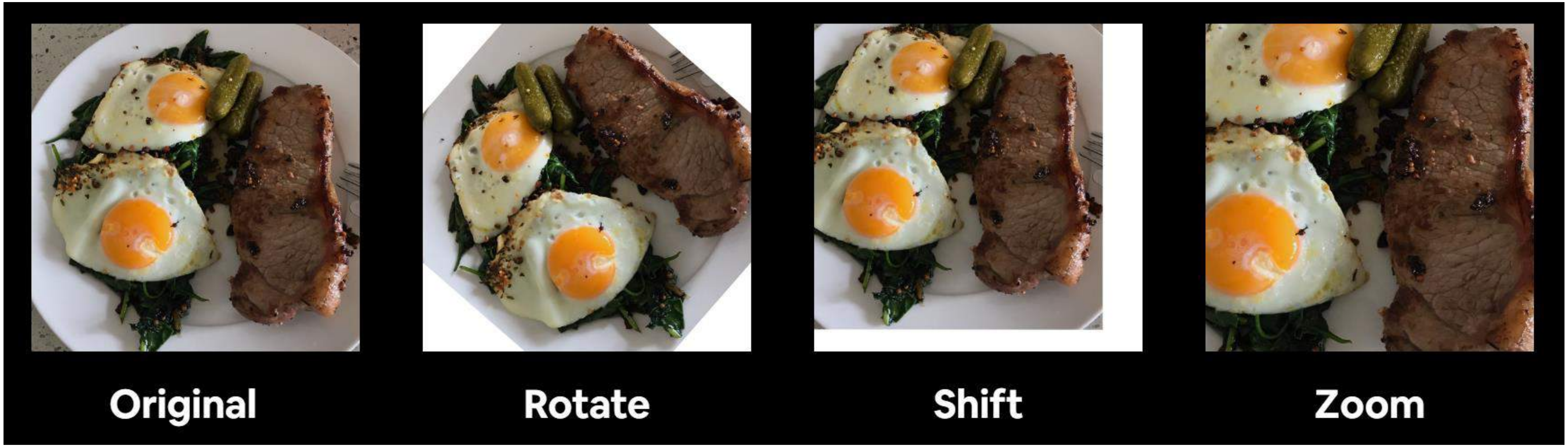


https://www.youtube.com/watch?v=pj9-rr1wDhM&t=263s

Analysing the MODEL


80% of training
10% of validation
10% of testing
80% of training
10% of validation
10% of testing
80% of training
10% of validation
10% of testing
200 sample
200 sample
200 sample
Well-fermented
Partly-brown
Partly-purple

Data Augmentation



Sensitivity Analysis

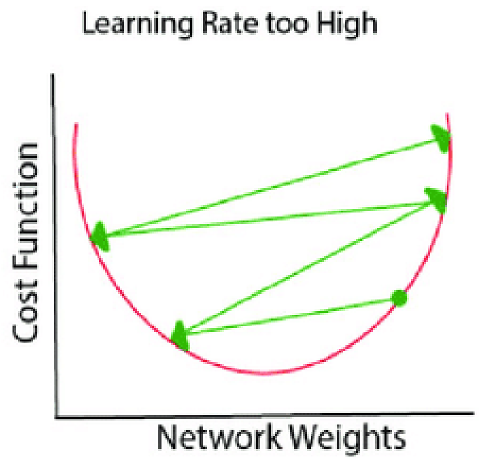
CNN Architecture	Testing accuracy							
	Dataset		Epoch		BS		LR	
	Unsegmented	Segmented	100	200	32	64	0.001	0.005
ResNet50	0.86	0.90	0.91	0.85	0.92	0.84	0.80	0.86
VGG16	0.33	0.40	0.33	0.40	0.33	0.40	0.40	0.33

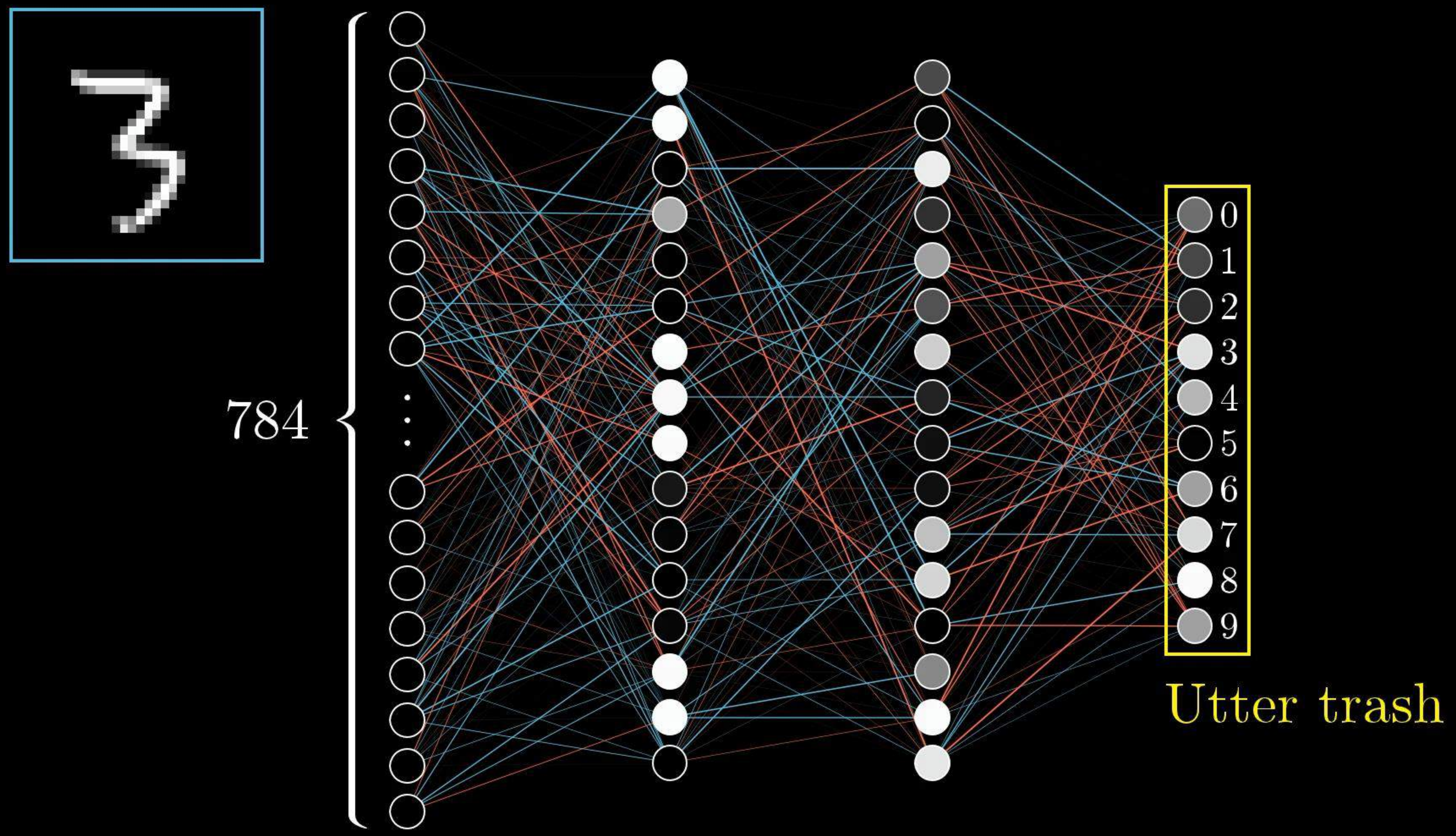
* BS: batch size; LR: learning rate; ResNet50: 50-layer residual neural network; VGG16: visual geometry group 16

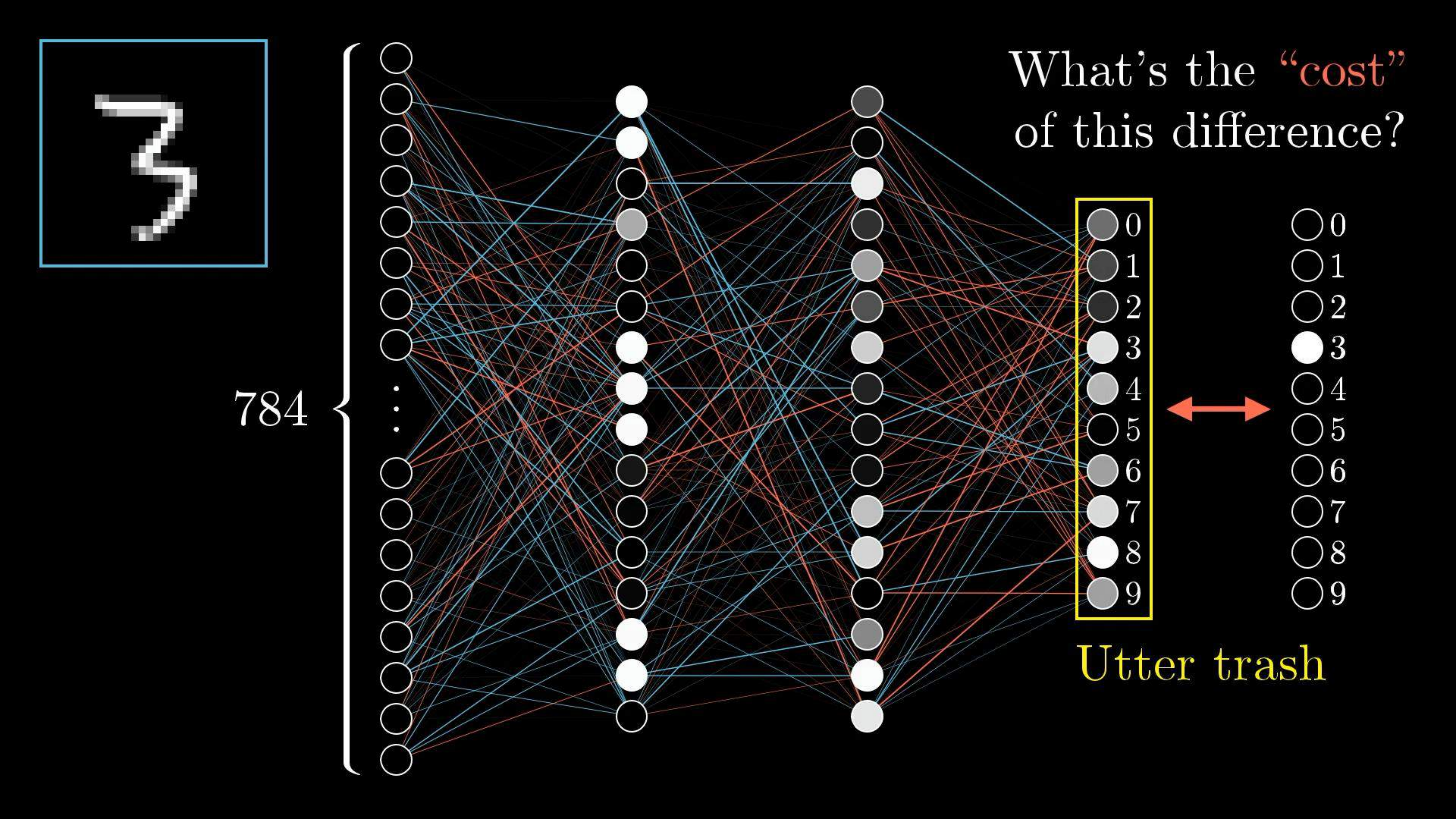
- Gambar disegmentasi = mengurangi noise/gangguan yang terdapat pada gambar (yang keberadaannya memungkinkan untuk dipelajari model dan dijadikan sebagai landasan untuk klasifikasi)
- **Higher epoch = semakin tinggi iterasi model dalam menelusuri data latih (training data)**
- **Higher batch size = Semakin banyak sampel gambar yang diproses/dipelajari model dalam tiap 1 iterasi**
- **Higher learning rate = semakin besar jangkauan step pembelajaran dalam mengupdate bobot parameter model selama optimisasi Ketika model ditraining**

Jika terdapat 600 gambar

- **32 batch size = minimal butuh 18.75 iterasi (epoch) untuk model mempelajari seluruh 600 gambar yang ada**
- **Jika model ditraining sebanyak 37.5 epoch = model mempelajari seluruh 600 gambar yang ada sebanyak 2x**



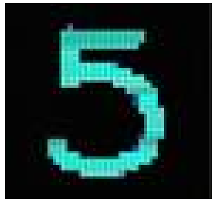

784
0
1
2
3
4
5
6
7
8
9
Utter trash

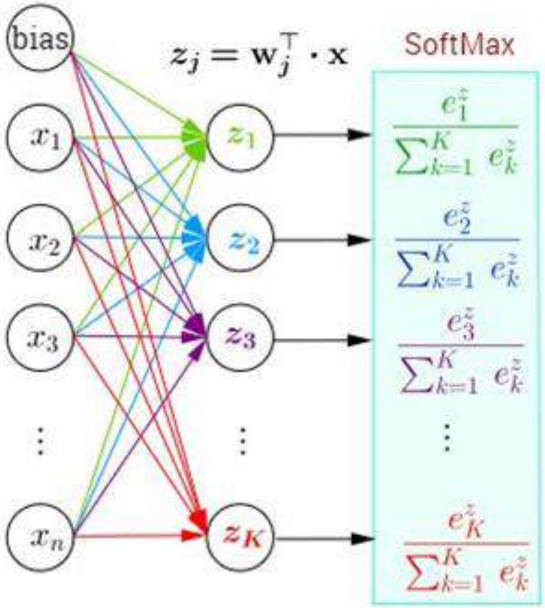


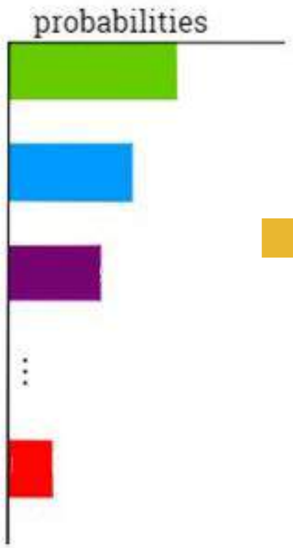
What's the "cost"
of this difference?
784
0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9
Utter trash

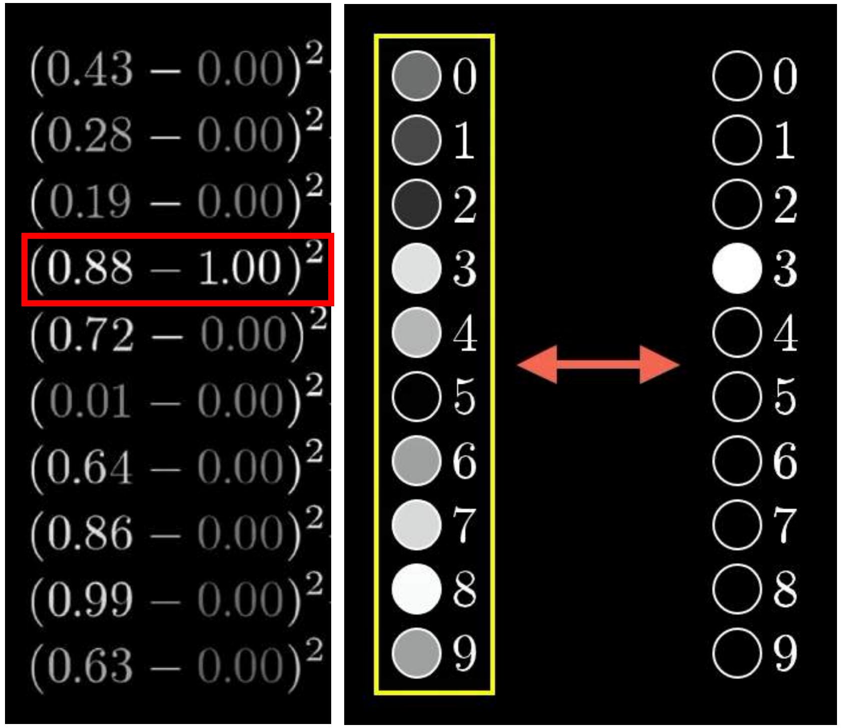
Klasifikasi Gambar Citra

Fungsi softmax akan melihat distribusi nilai probabilitas masing-masing output









Model loss

Cost model dihitung dengan menghitung rata-rata loss yang didapatkan

Semakin tinggi cost-nya = prediksi yang dilakukan kurang efisien dan akurasi masih kurang maksimal

$$\boxed{3.37}\left\{\begin{array}{lcl} 0.1863 & \leftarrow & (0.43-0.00)^2+ \\ 0.0809 & \leftarrow & (0.28-0.00)^2+ \\ 0.0357 & \leftarrow & (0.19-0.00)^2+ \\ 0.0138 & \leftarrow & \boxed{(0.88-1.00)^2}+ \\ 0.5242 & \leftarrow & (0.72-0.00)^2+ \\ 0.0001 & \leftarrow & (0.01-0.00)^2+ \\ 0.4079 & \leftarrow & (0.64-0.00)^2+ \\ 0.7388 & \leftarrow & (0.86-0.00)^2+ \\ 0.9817 & \leftarrow & (0.99-0.00)^2+ \\ 0.3998 & \leftarrow & (0.63-0.00)^2 \end{array}\right.$$

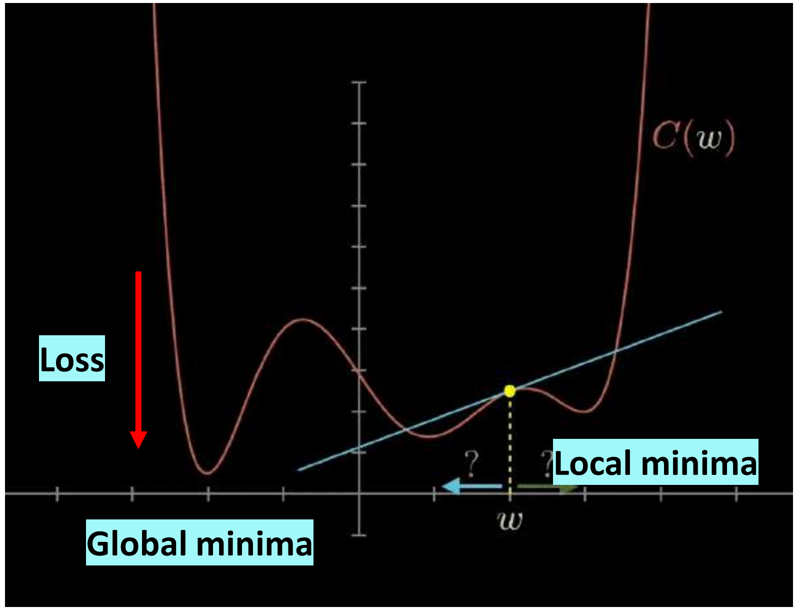
Kurang maksimal

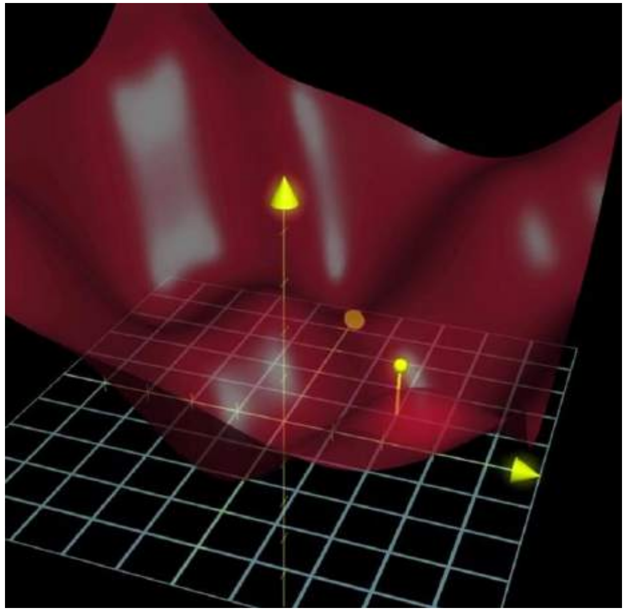
$$\boxed{0.03}\left\{\begin{array}{lcl} 0.0006 & \leftarrow & (0.02-0.00)^2+ \\ 0.0007 & \leftarrow & (0.03-0.00)^2+ \\ 0.0039 & \leftarrow & (0.06-0.00)^2+ \\ 0.0009 & \leftarrow & \boxed{(0.97-1.00)^2}+ \\ 0.0055 & \leftarrow & (0.07-0.00)^2+ \\ 0.0004 & \leftarrow & (0.02-0.00)^2+ \\ 0.0022 & \leftarrow & (0.05-0.00)^2+ \\ 0.0033 & \leftarrow & (0.06-0.00)^2+ \\ 0.0072 & \leftarrow & (0.08-0.00)^2+ \\ 0.0018 & \leftarrow & (0.04-0.00)^2 \end{array}\right.$$

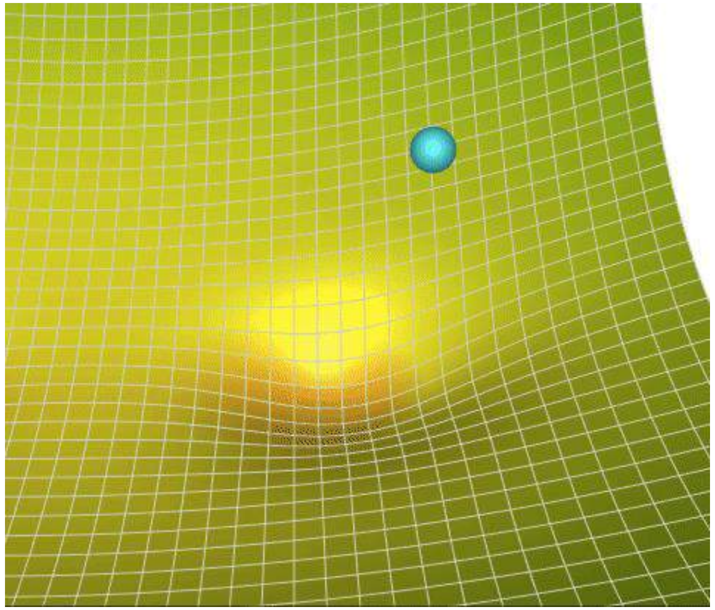
Maksimal

Optimization

Selama optimasi model, nilai Loss yang dihasilkan tergantung pada jangkauan step learning rate
1x running training (1 epoch) = bergerak 1 step

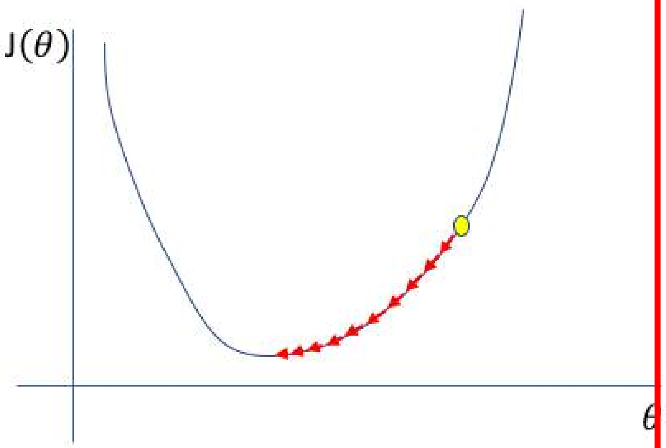






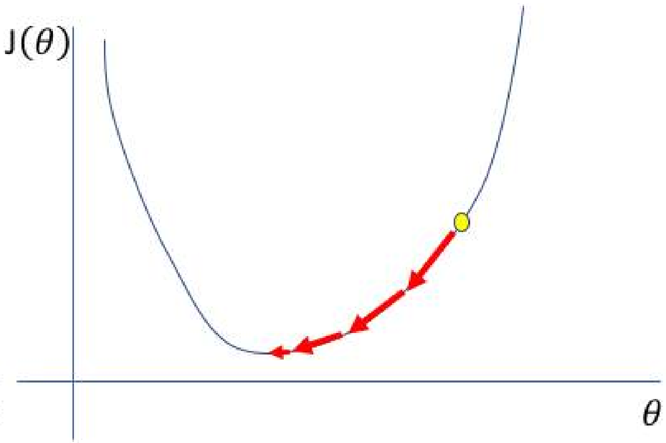
https://www.3blue1brown.com/lessons/gradient-descent
https://youtu.be/IHZwWFHWa-w

Too low



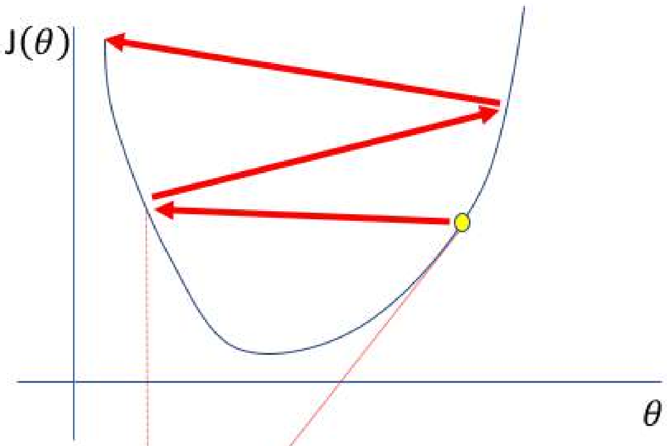
A small learning rate requires many updates before reaching the minimum point

Just right

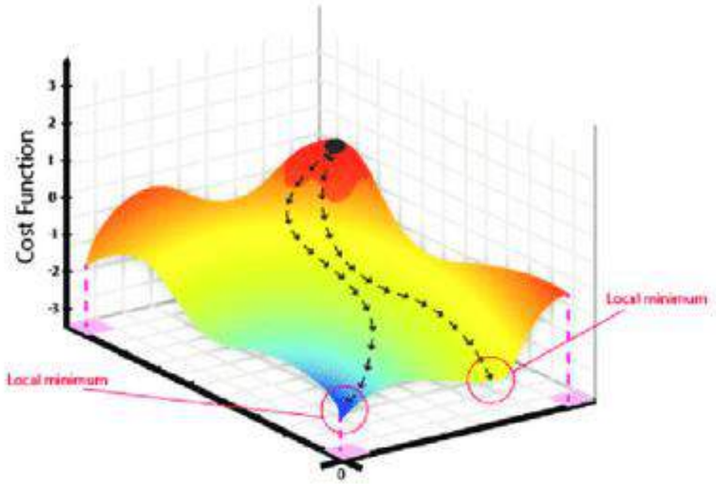


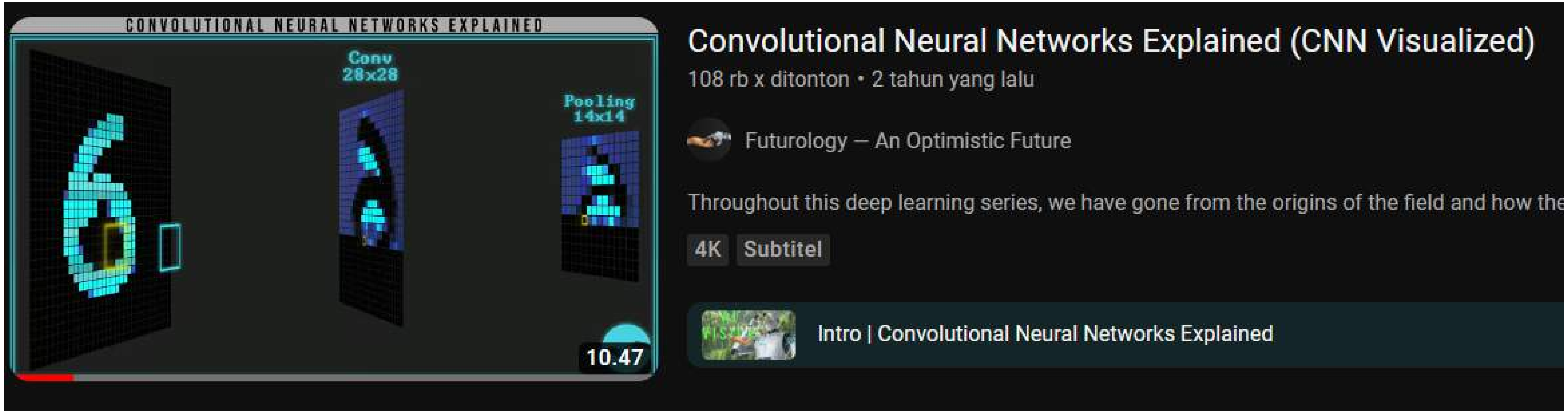
The optimal learning rate swiftly reaches the minimum point

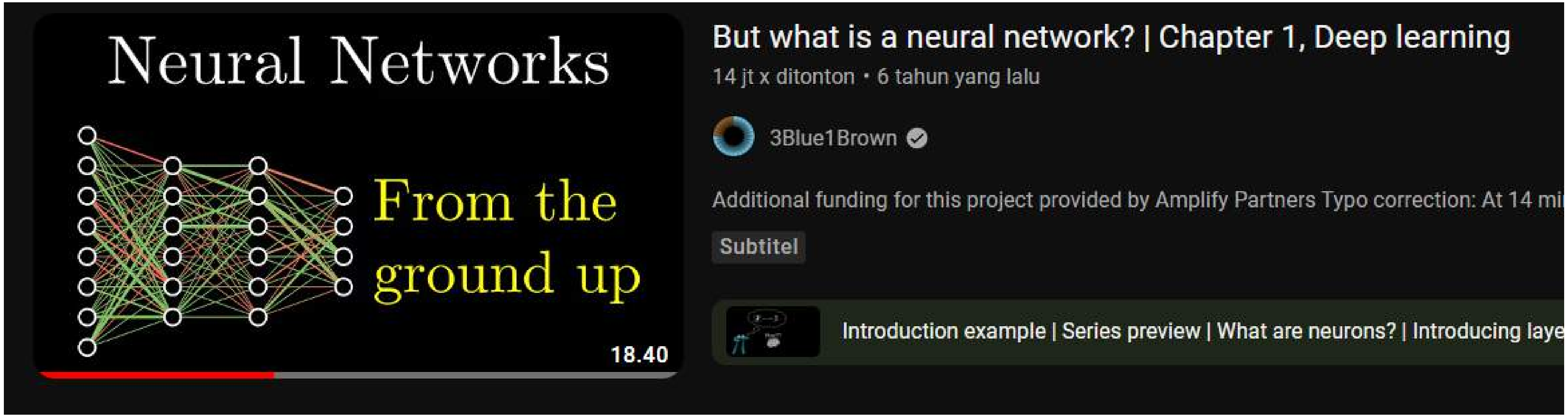
Too high



Too large of a learning rate causes drastic updates which lead to divergent behaviors




CONVOLUTIONAL NEURAL NETWORKS EXPLAINED
Conv
28x28
Pooling
14x14
10.47
Convolutional Neural Networks Explained (CNN Visualized)
108 rb x ditonton • 2 tahun yang lalu
Futurology — An Optimistic Future
Throughout this deep learning series, we have gone from the origins of the field and how the
4K
Subtitel
Intro | Convolutional Neural Networks Explained


Neural Networks
From the
ground up
18.40
But what is a neural network? | Chapter 1, Deep learning
14 jt x ditonton • 6 tahun yang lalu
3Blue1Brown
Additional funding for this project provided by Amplify Partners Typo correction: At 14 mi
Subtitel
Introduction example | Series preview | What are neurons? | Introducing laye