Deconvoluting Convolutions

DATA SCIENCE MEETUP

APRIL 2018

Topics

The 2018 Data Science Bowl

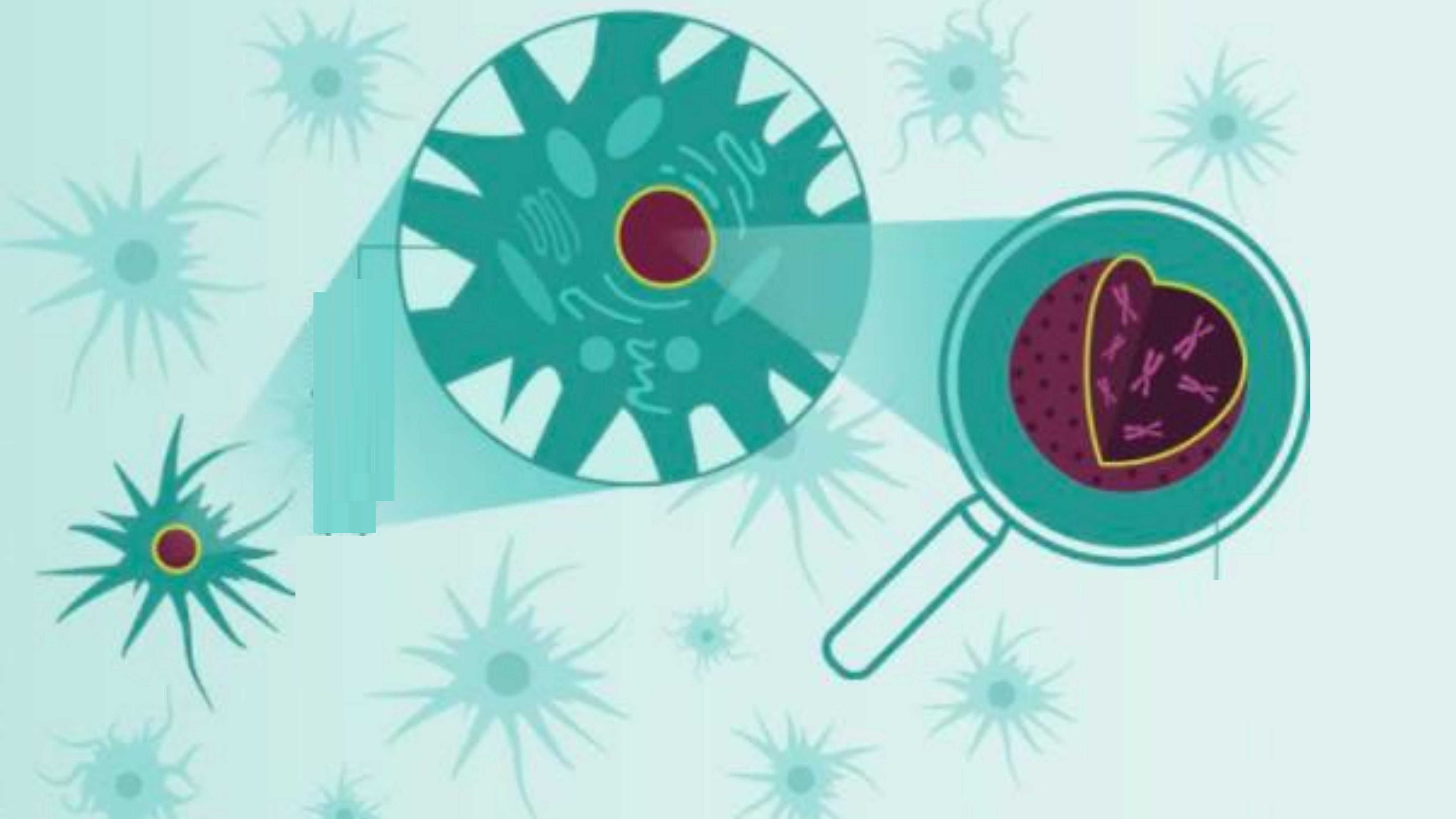
Puppy bowl

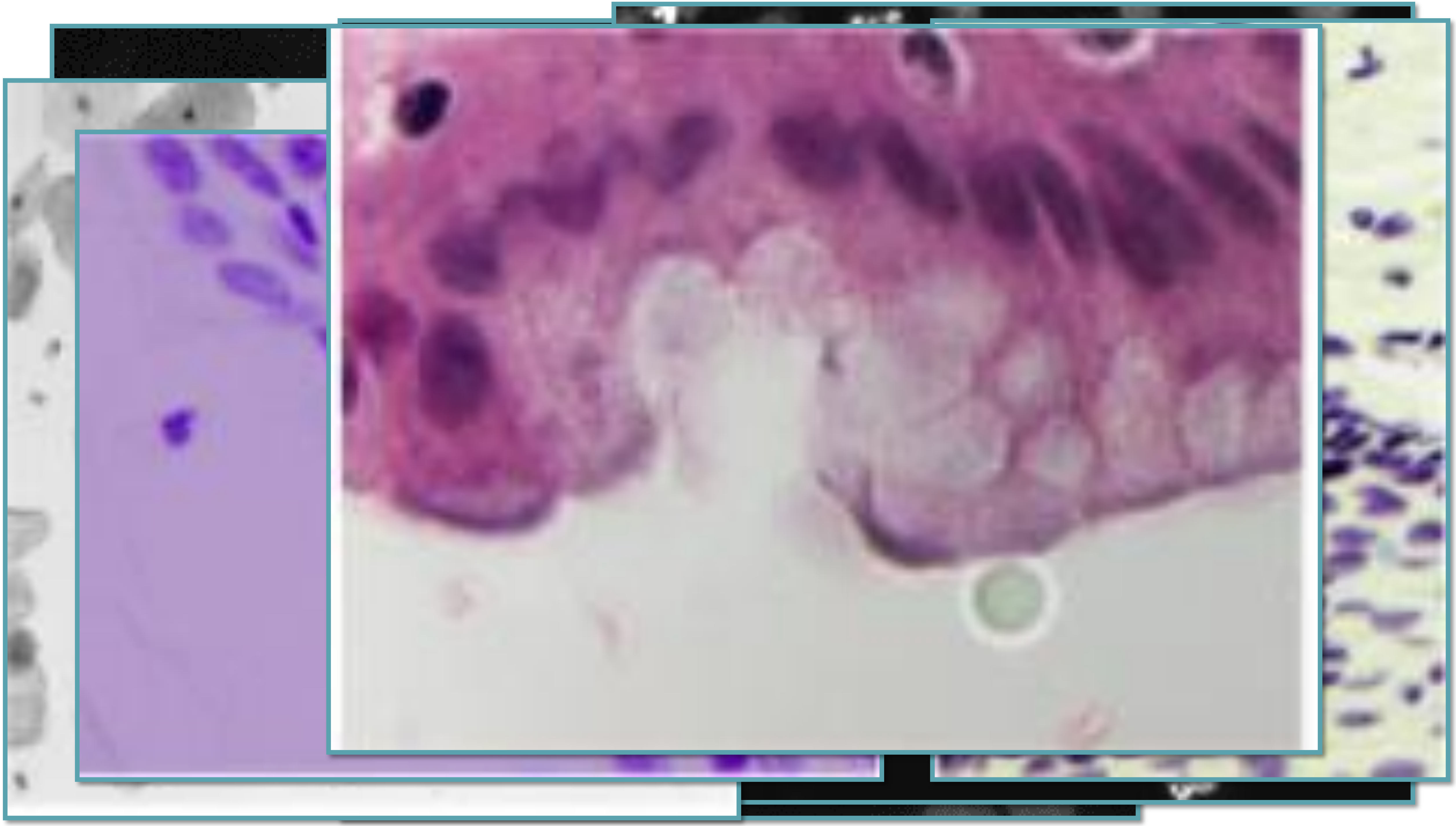
- Image processing basics
- The simple approach

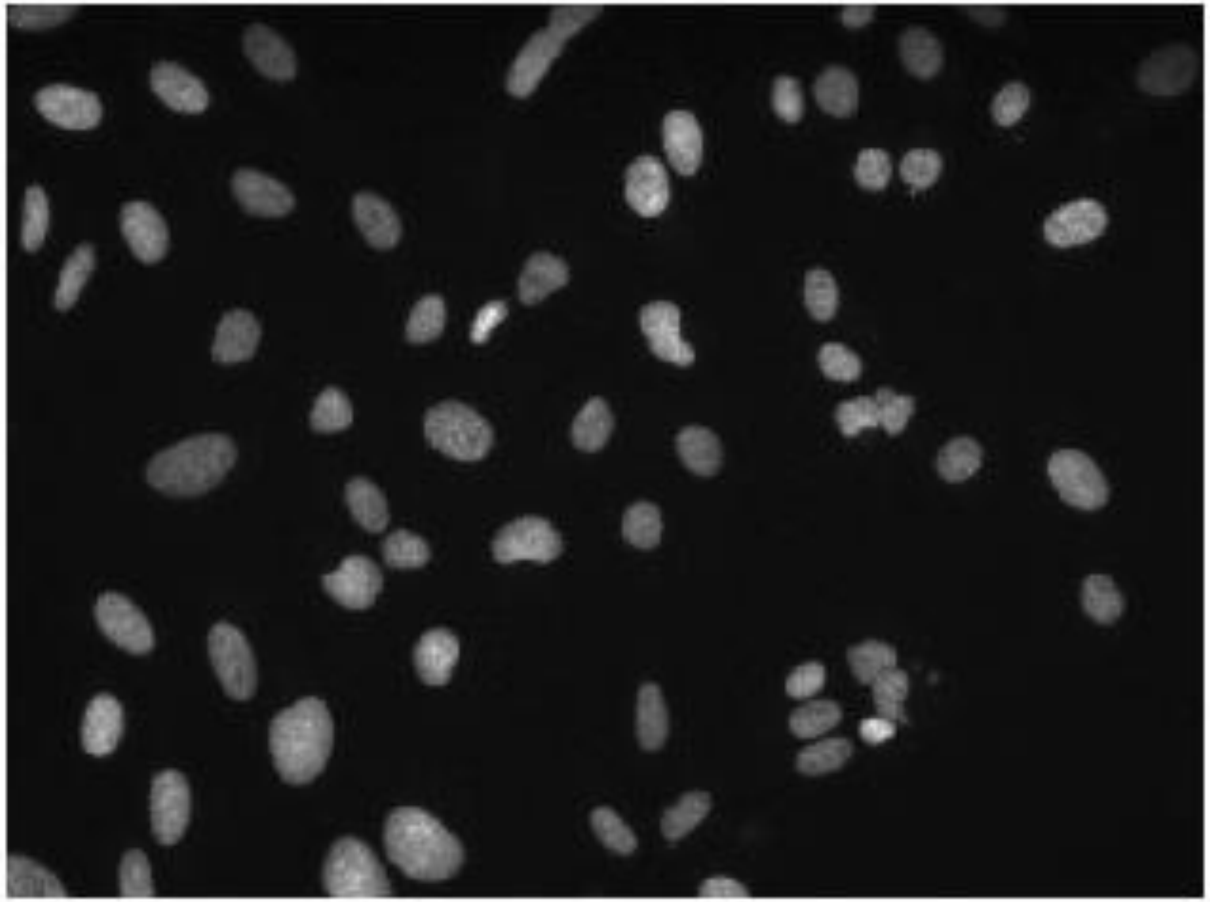
Superbowl

- Feature detection basics
- The U-Net approach

Performance

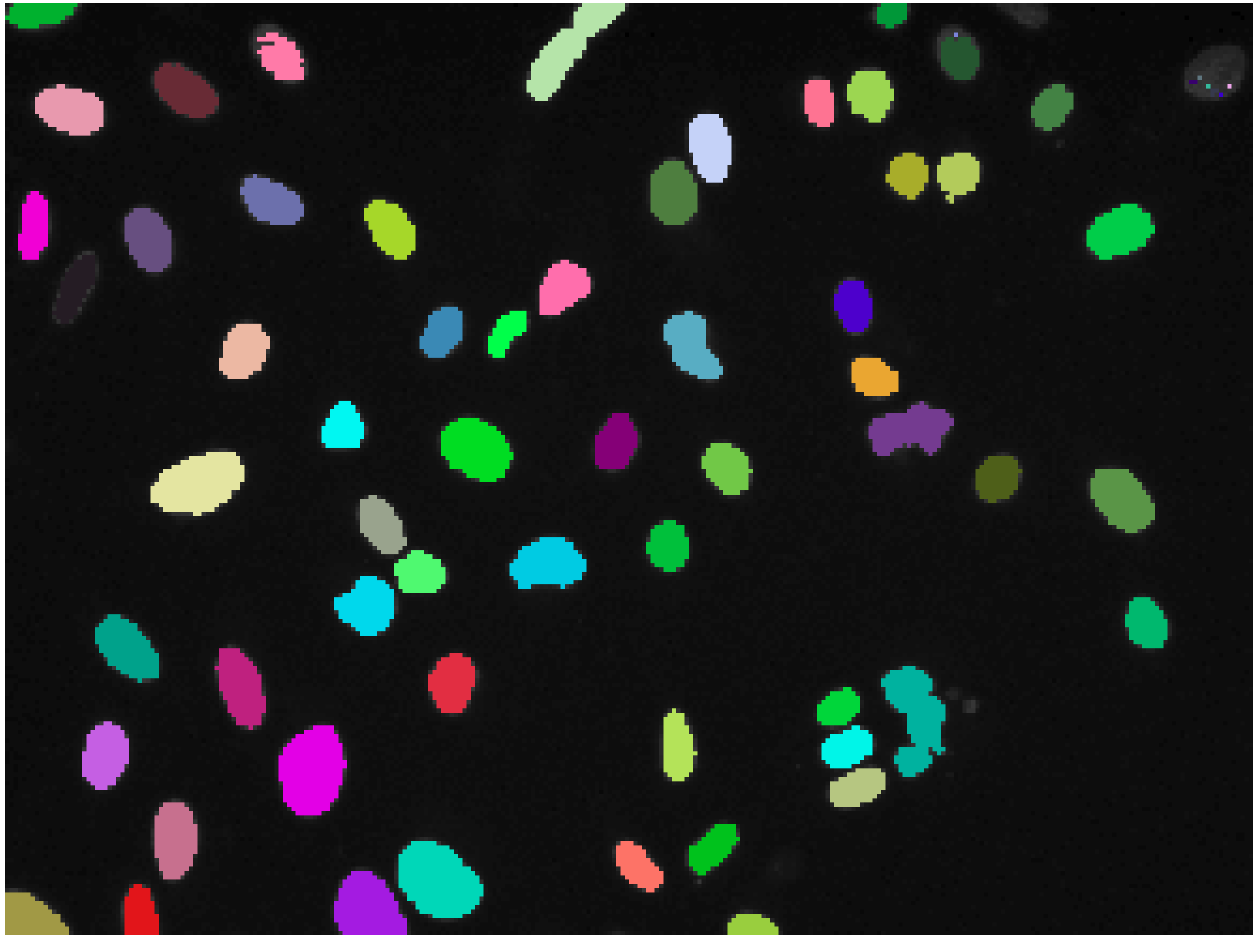






Competition goal:

Separate mask for each nucleus



Competition metric:

Intersection of the Union

Competition challenges

Variety of colors, microscopy types

Variety of sizes and scales

Different background tissue / nuclei types

Overlapping nuclei

Small training set (~700)

Errors in training set

Topics

The 2018 Data Science Bowl

Puppy bowl

- **Image processing basics**
- The simple approach

Superbowl

- Feature detection basics
- The U-Net approach

Performance

Image:
Intensity data with meaningful order

One cell from train image

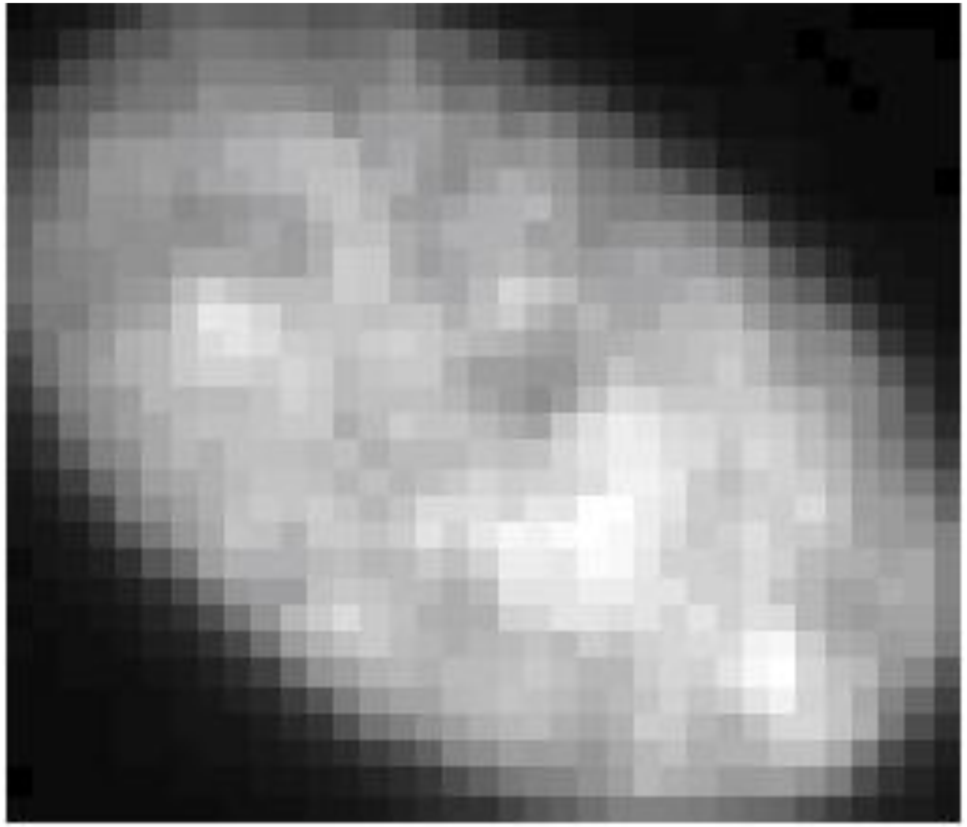
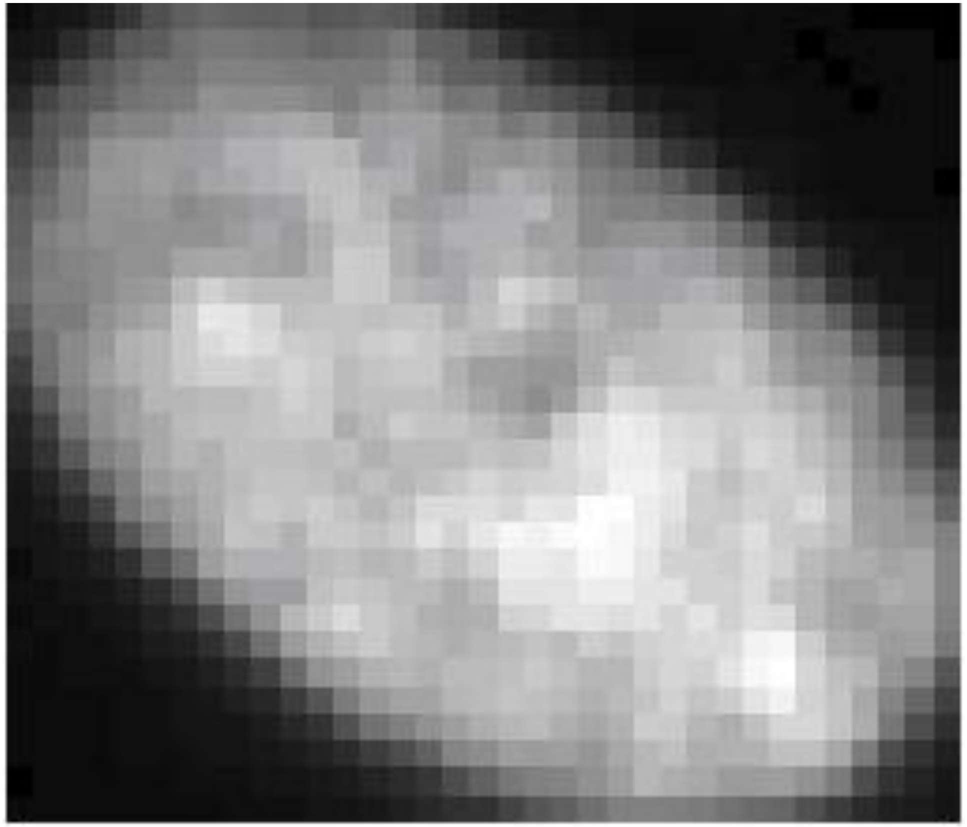


Image:
Intensity data with meaningful order

One cell from train image



Data with 3 color channels

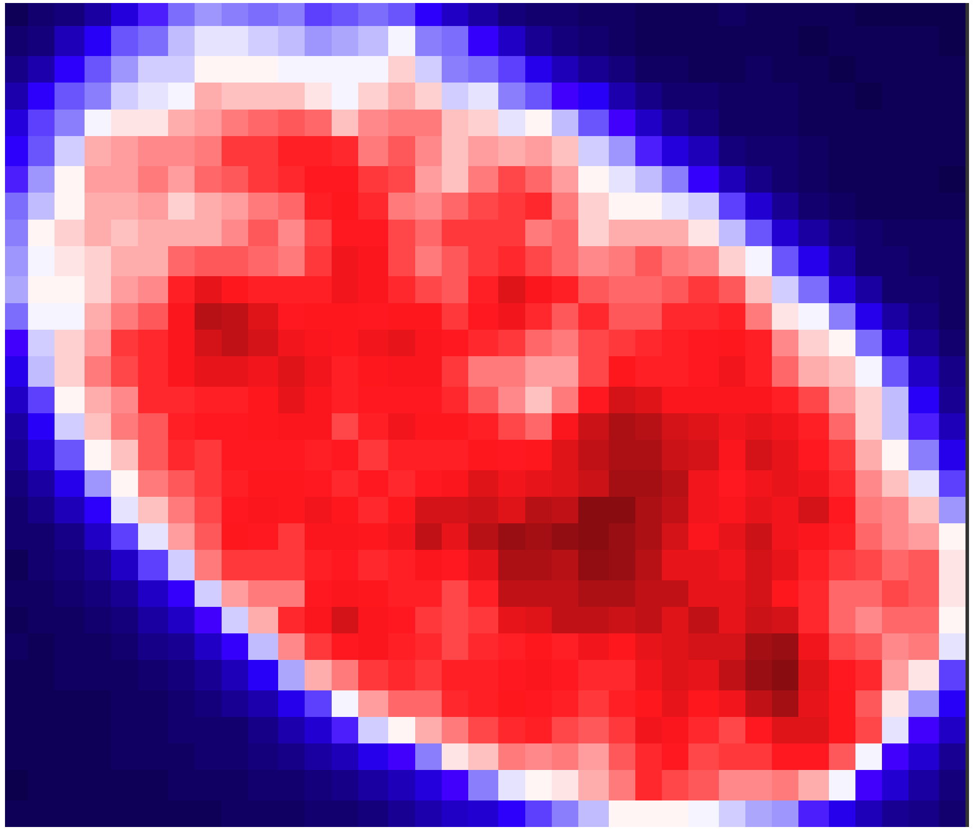
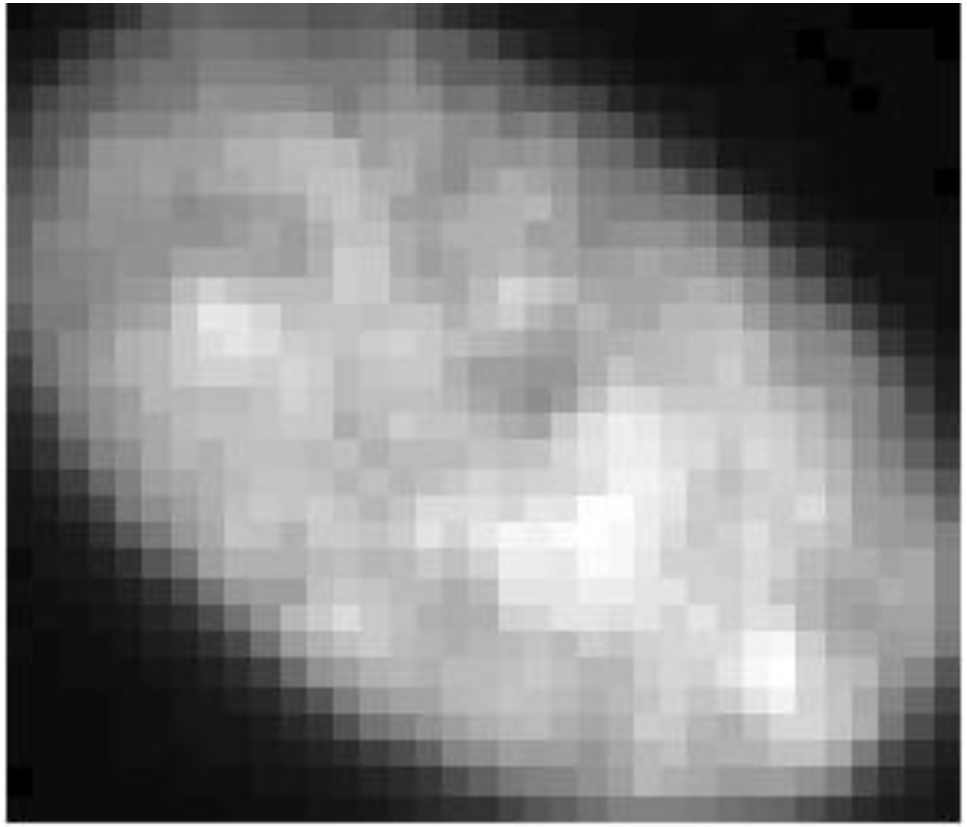
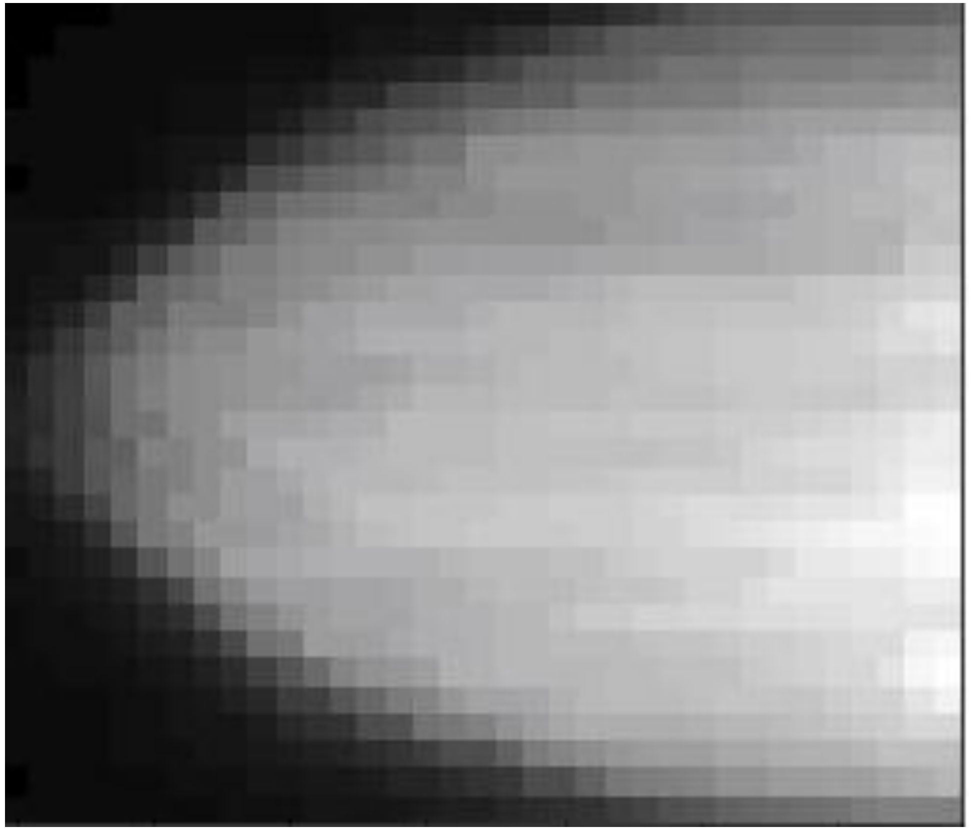


Image:
Intensity data **with meaningful order**

One cell from train image



Data sorted along rows



Topics

The 2018 Data Science Bowl

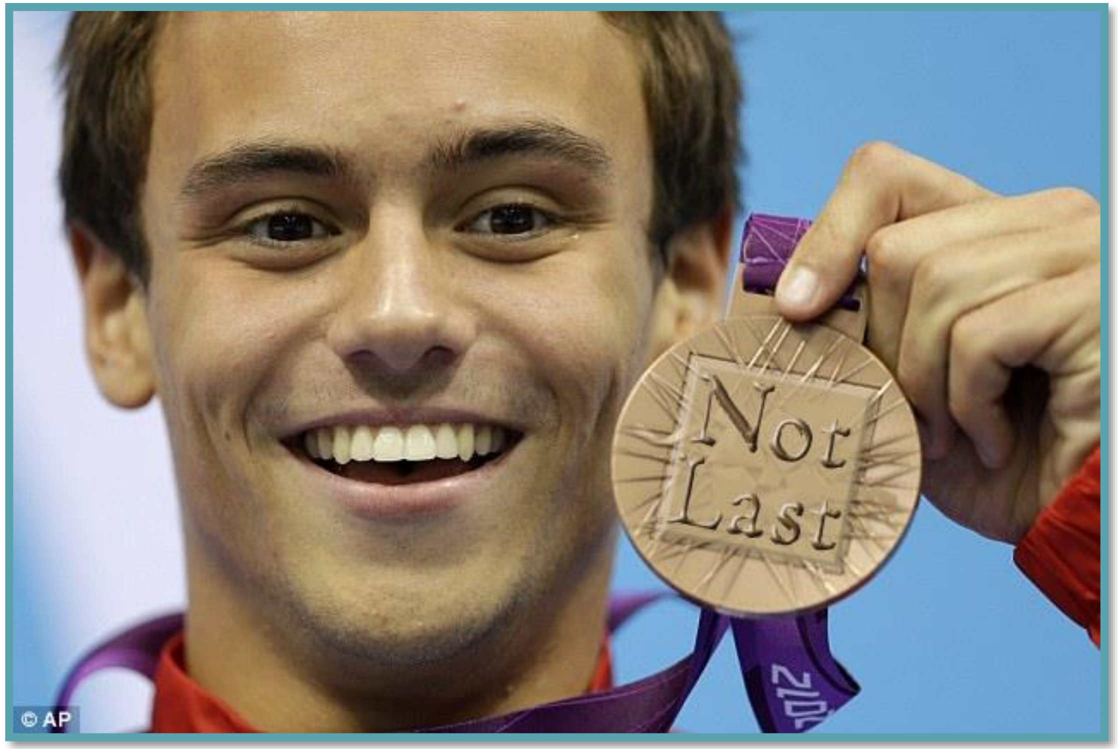
Puppy bowl

- Image processing basics
- **The simple approach**

Superbowl

- Feature detection basics
- The U-Net approach

Performance



kaggle Search kaggle Competitions Datasets Kernels Discussion Learn •••

Stephen Bailey

Teaching notebook for total imaging newbies

352 voters

last run 2 months ago · Python notebook · 18866 views
using data from 2018 Data Science Bowl · Public

Notebook Code Data (1) Output (1) Comments (46) Log Versions (14) Forks (118) Options Fork Notebook Edit No

Tags Add Tag

Notebook

This kernel will implement classical image techniques and will hopefully serve as a useful primer to people who have never worked with image data before. Ultimately, we will develop a simple pipeline using `scipy` and `numpy` (and a little bit of

Competition challenges

Small training set (~700)

Variety of colors, microscopy types

Variety of sizes and scales

Different background tissue

Overlapping nuclei

Competition challenges

~~Small training set (~700)~~

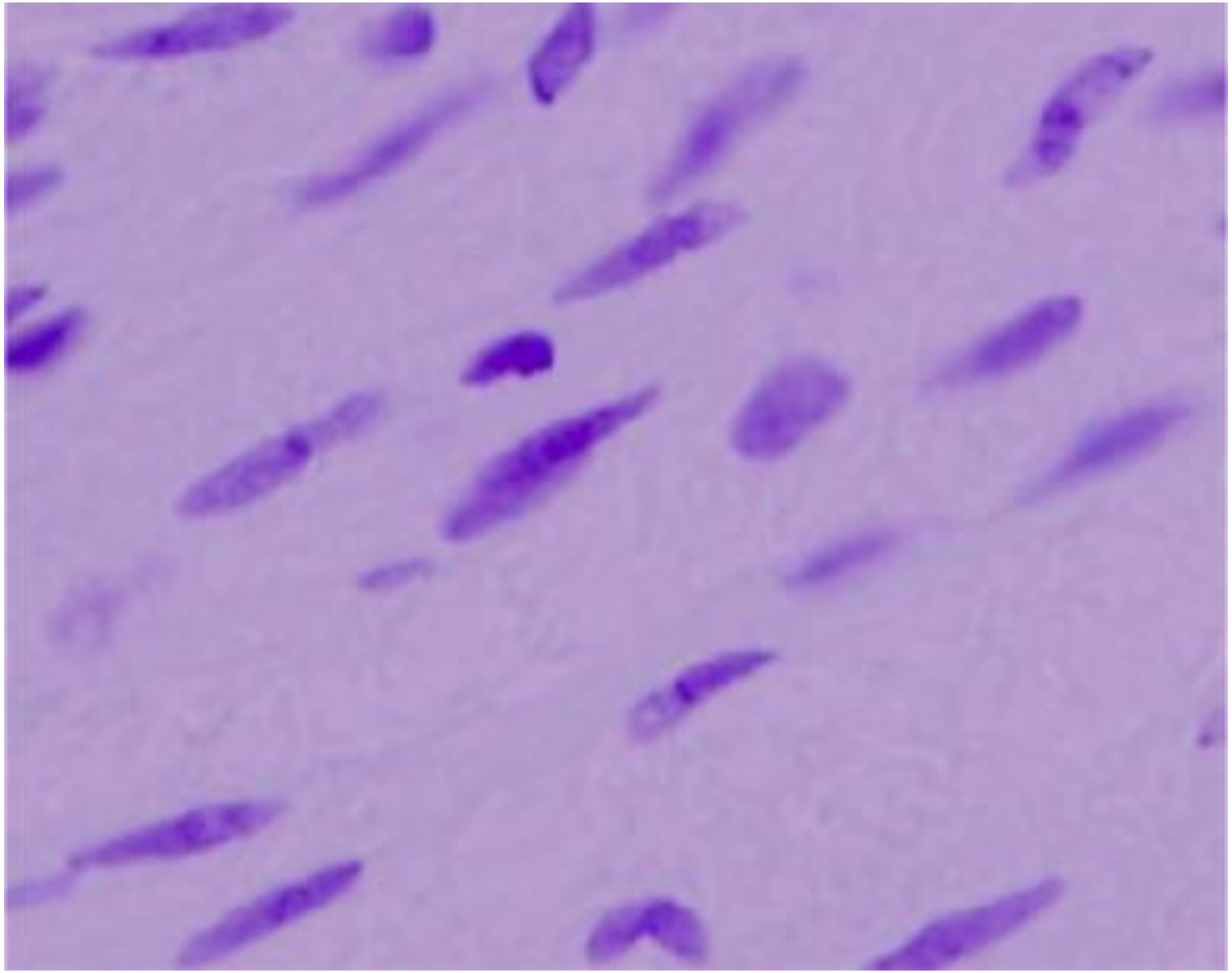
~~Variety of colors, microscopy types~~

~~Variety of sizes and scales~~

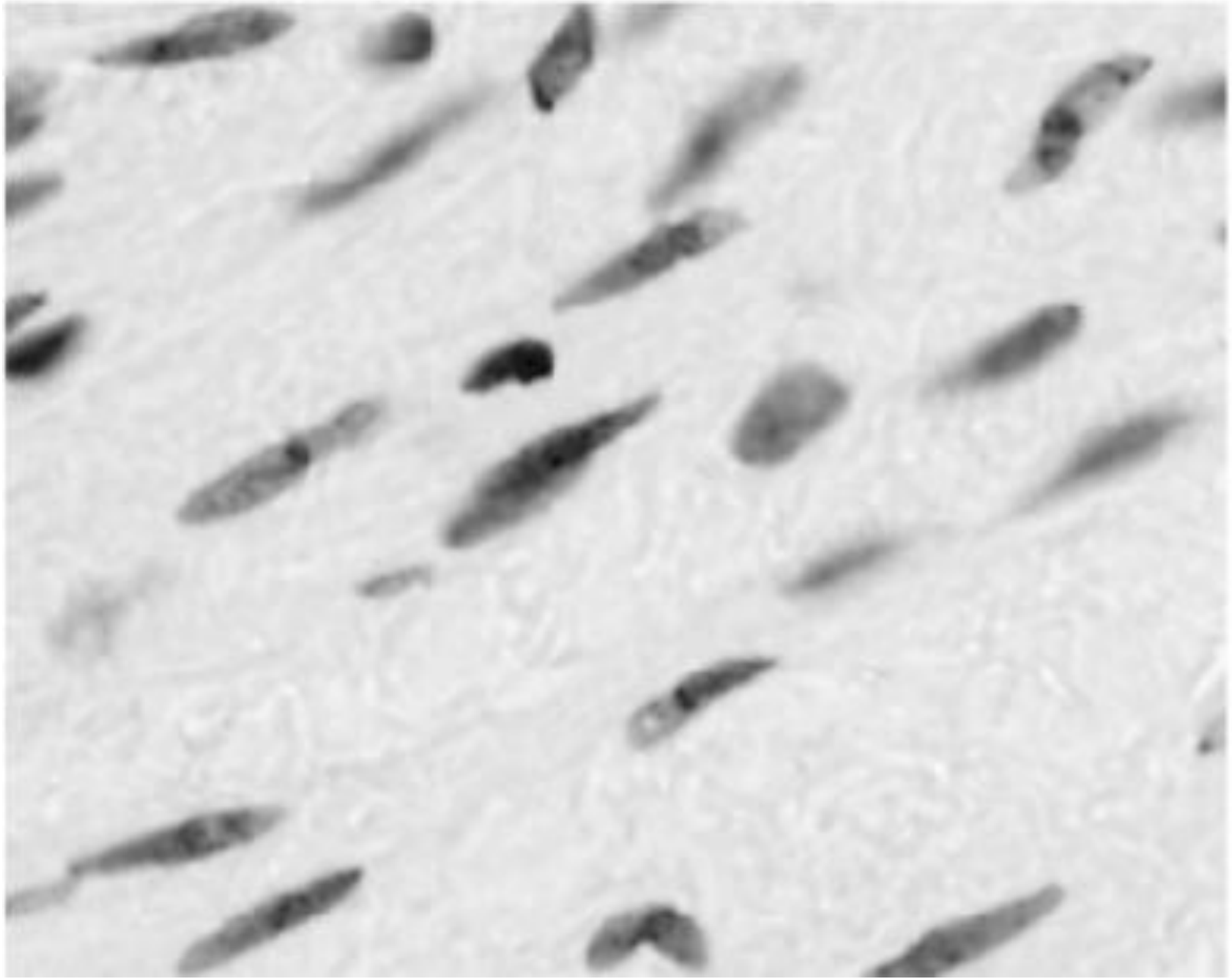
~~Different background tissue~~

~~Overlapping nuclei~~

Step one: Convert to grayscale image.



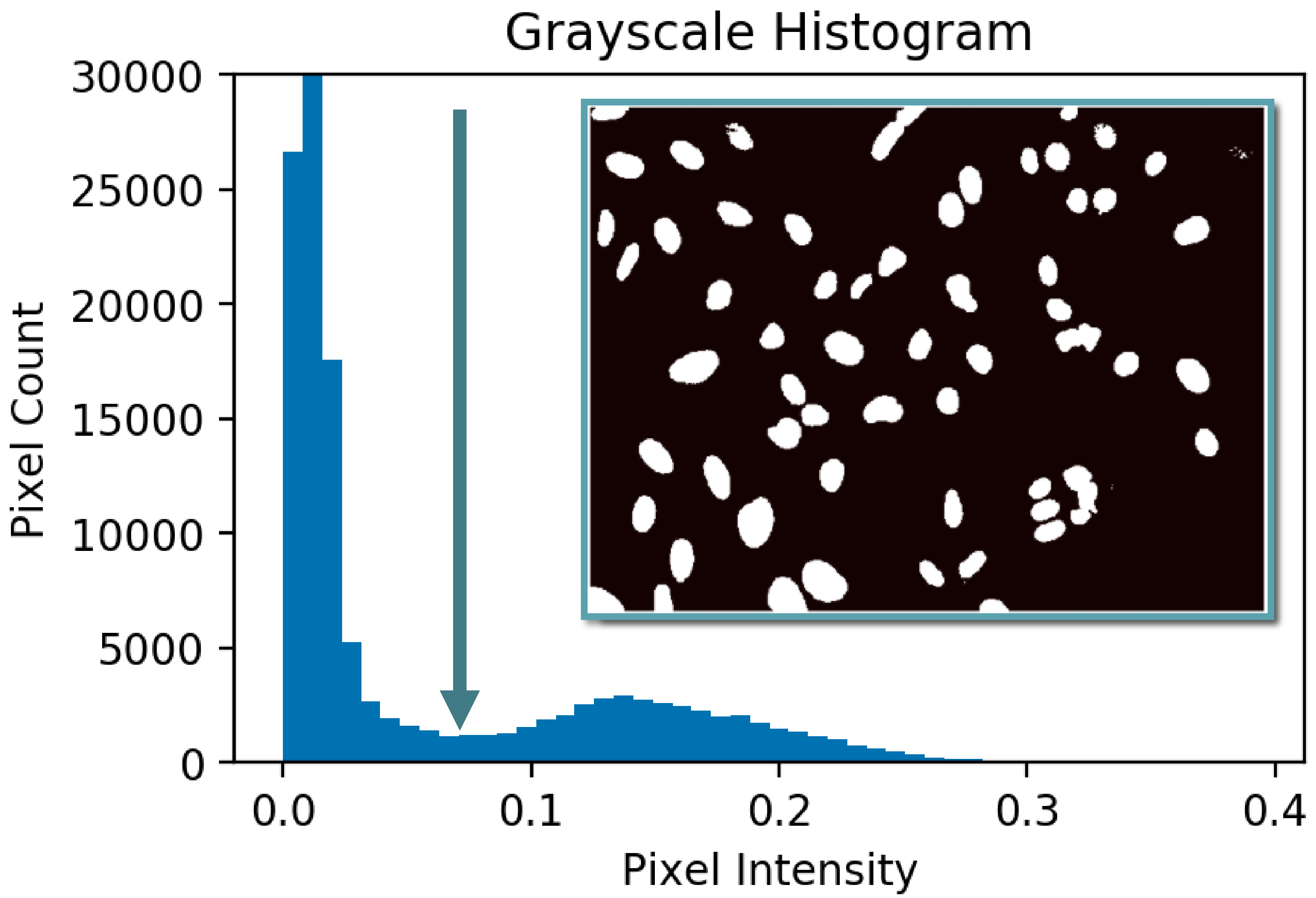
Step one: Convert to grayscale image.



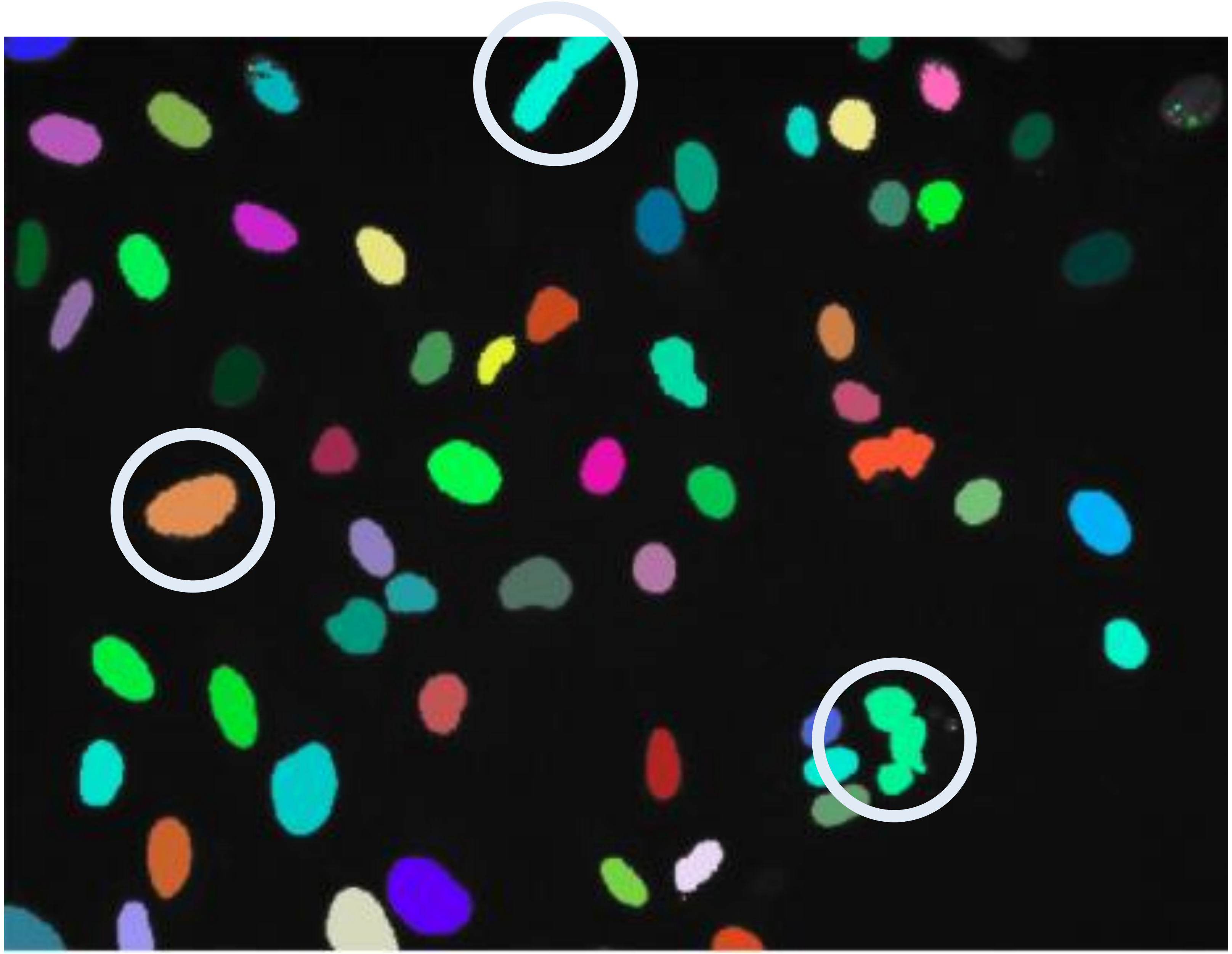
Step two: Threshold background

Assume there are two pixel types in the image:

- Nuclei
- Background.

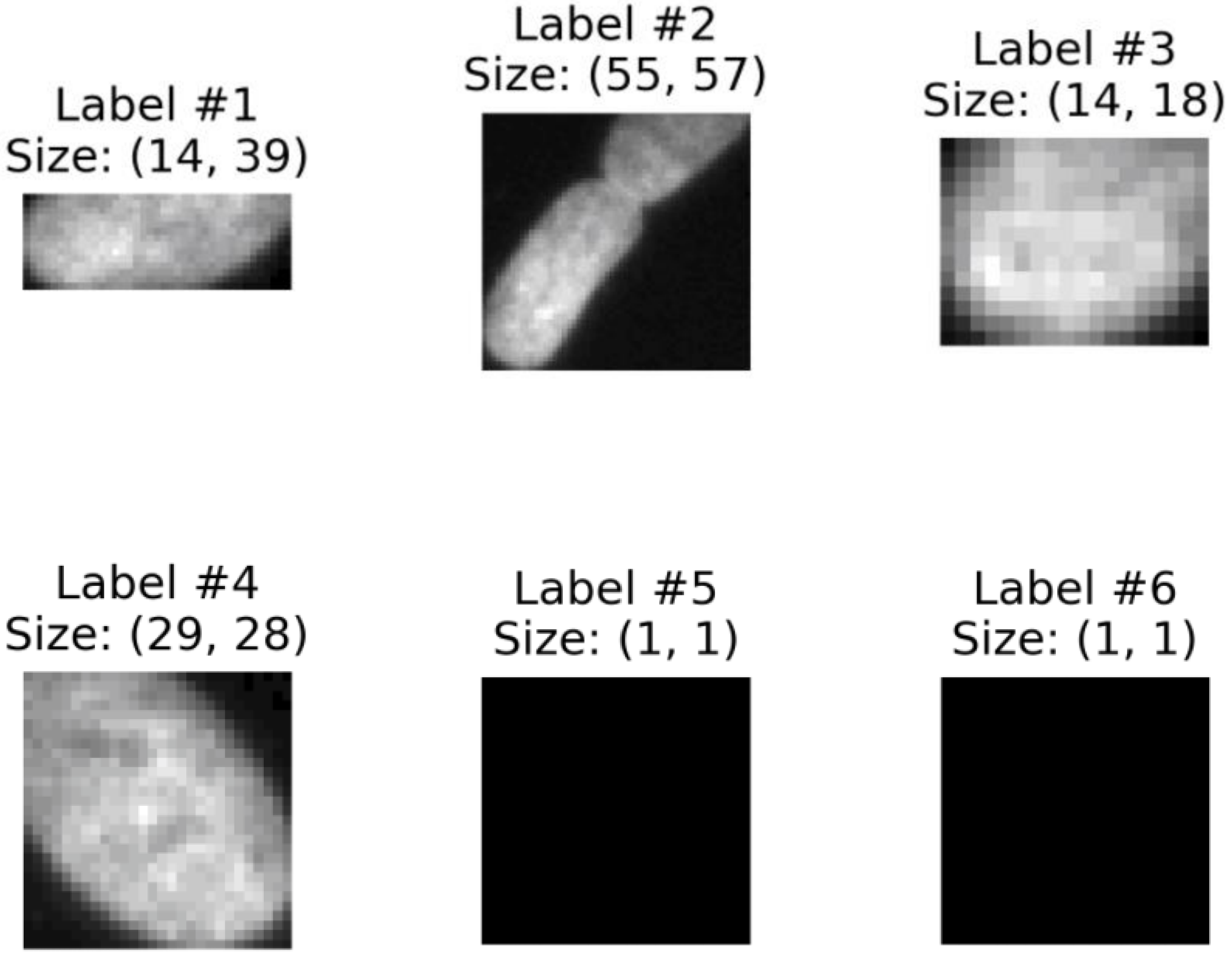


Step three: Separate individual objects



Step four: Post-process individual objects

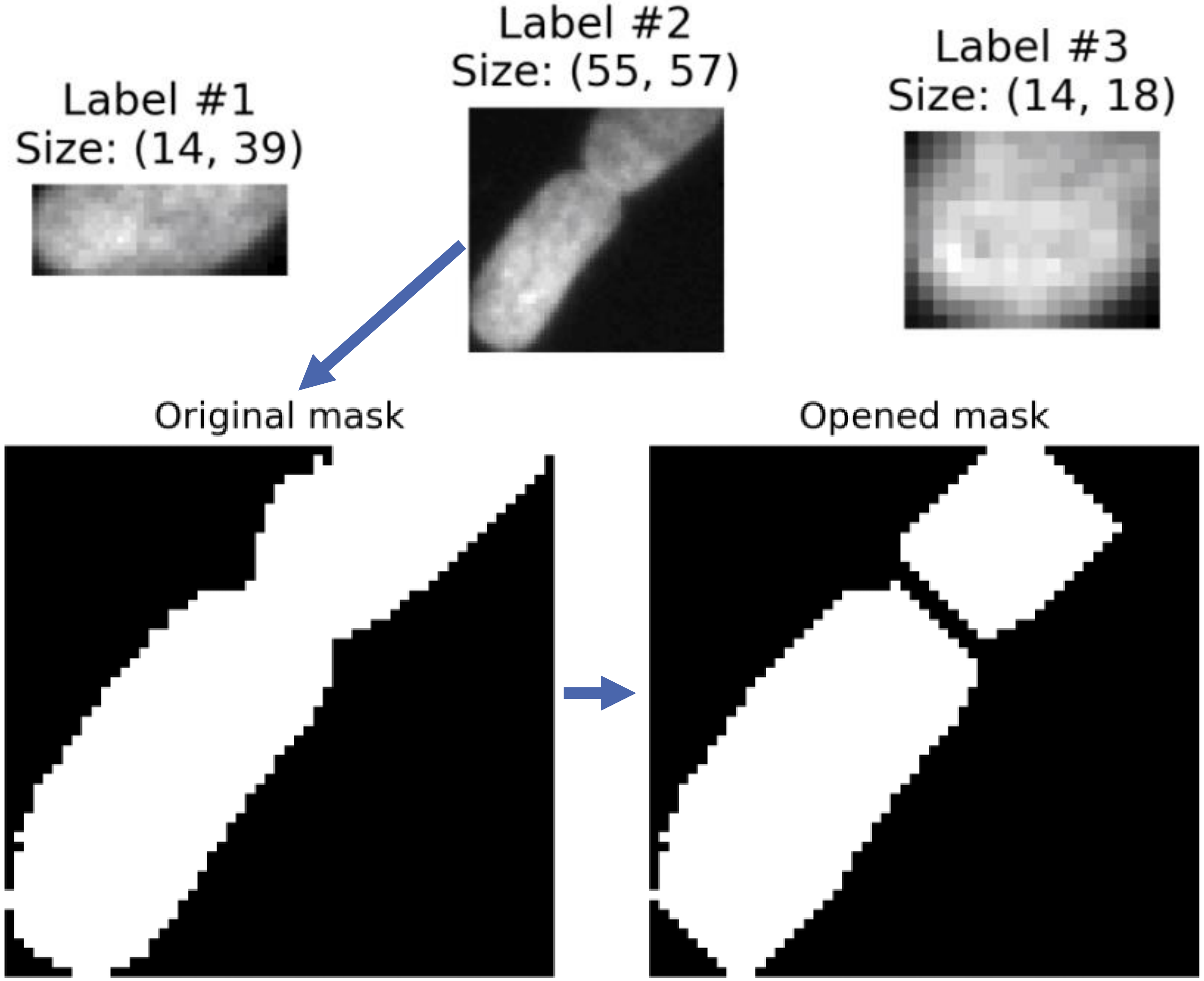
Eliminate tiny objects



Step four: Post-process individual objects

Eliminate tiny objects

Open connected objects




WE NEED TO GO
DEEPER

Topics

The 2018 Data Science Bowl

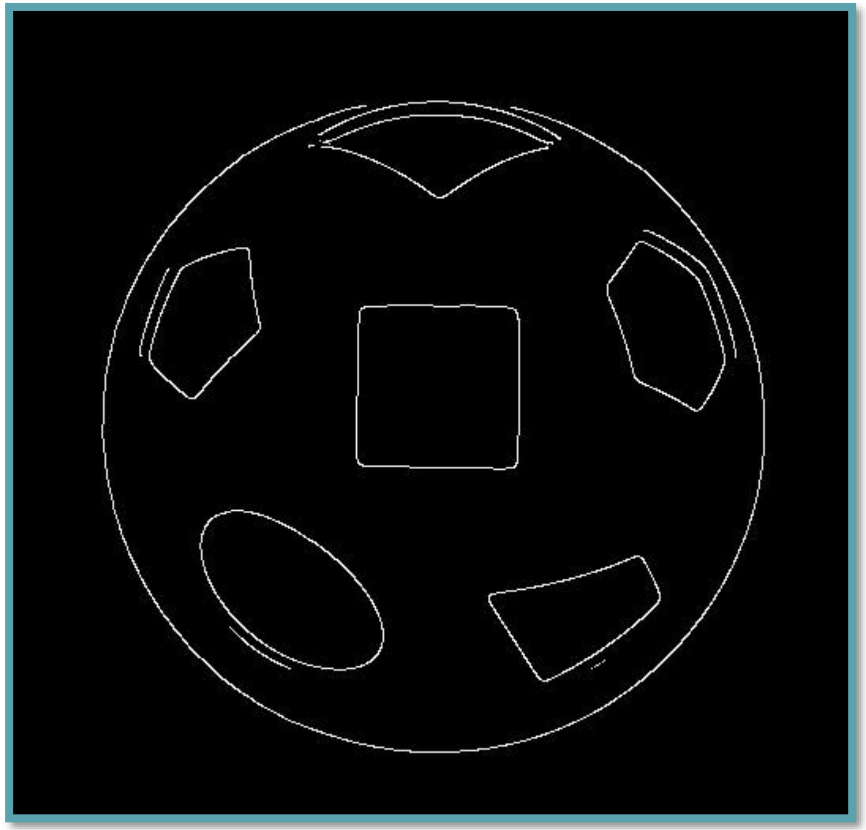
Puppy bowl

- Image processing basics
- The simple approach

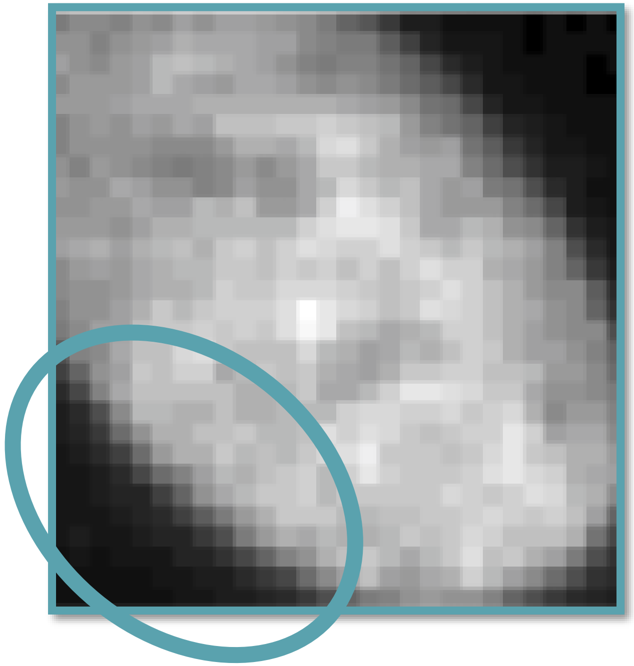
Superbowl

- **Feature detection basics**
- The U-Net approach

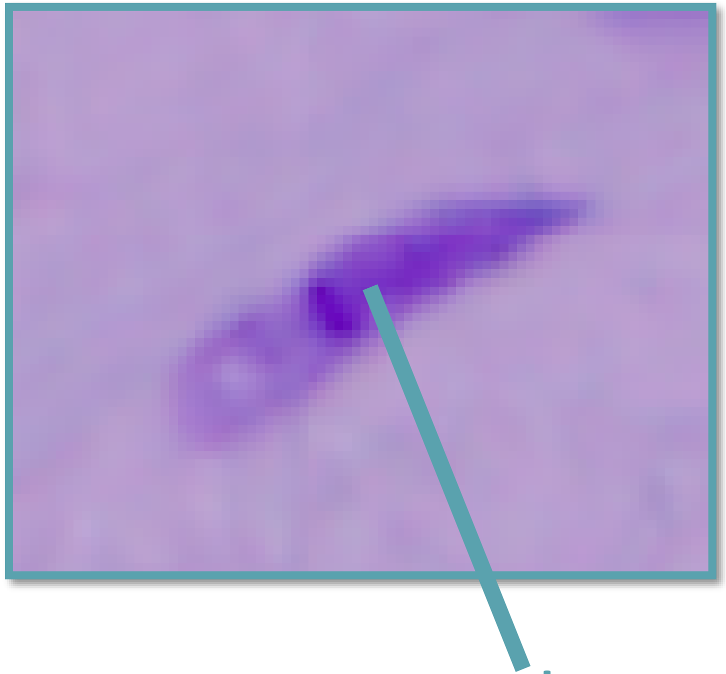
Performance



Humans recognize objects from patterns



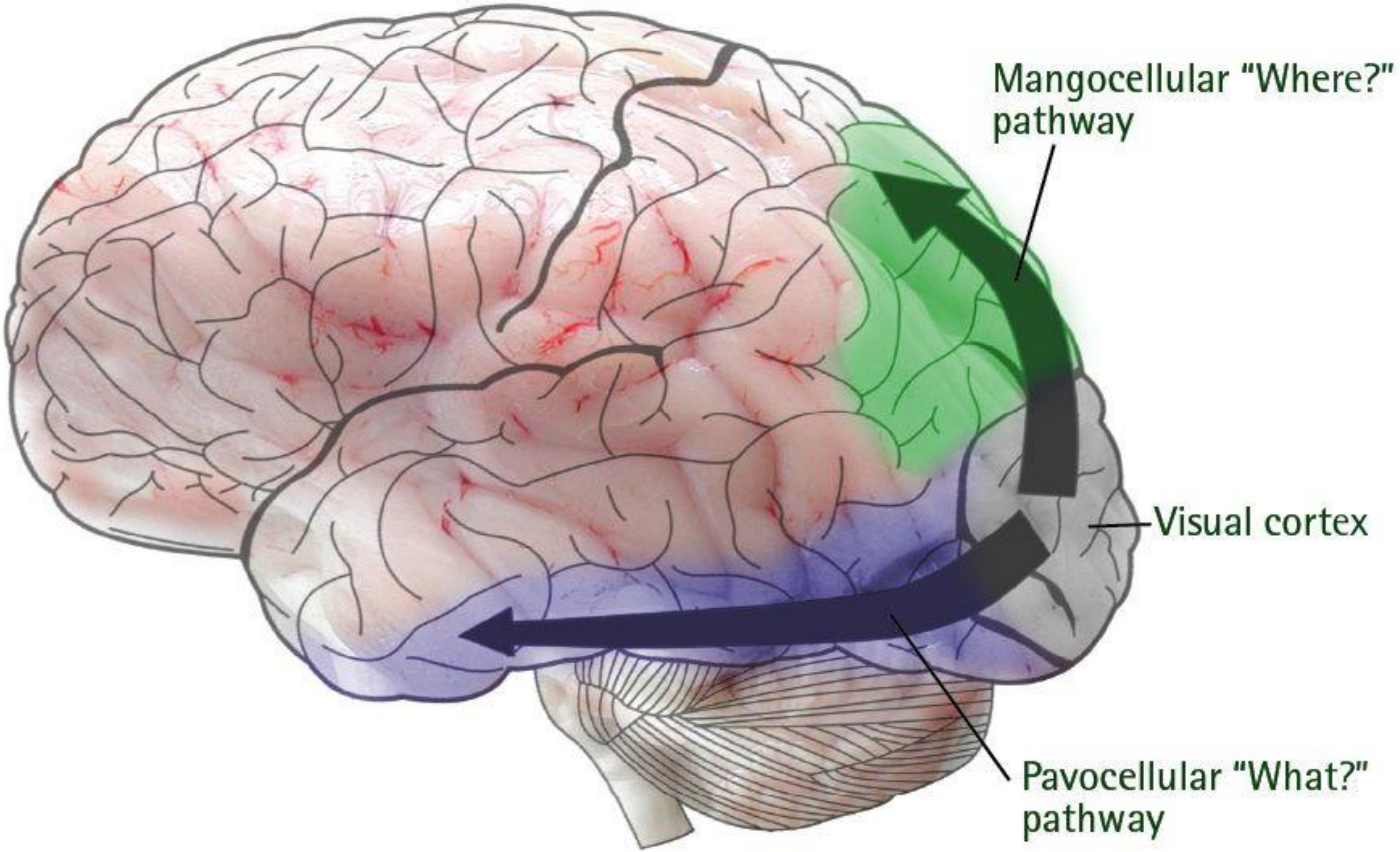
Edges



Color

Ratios

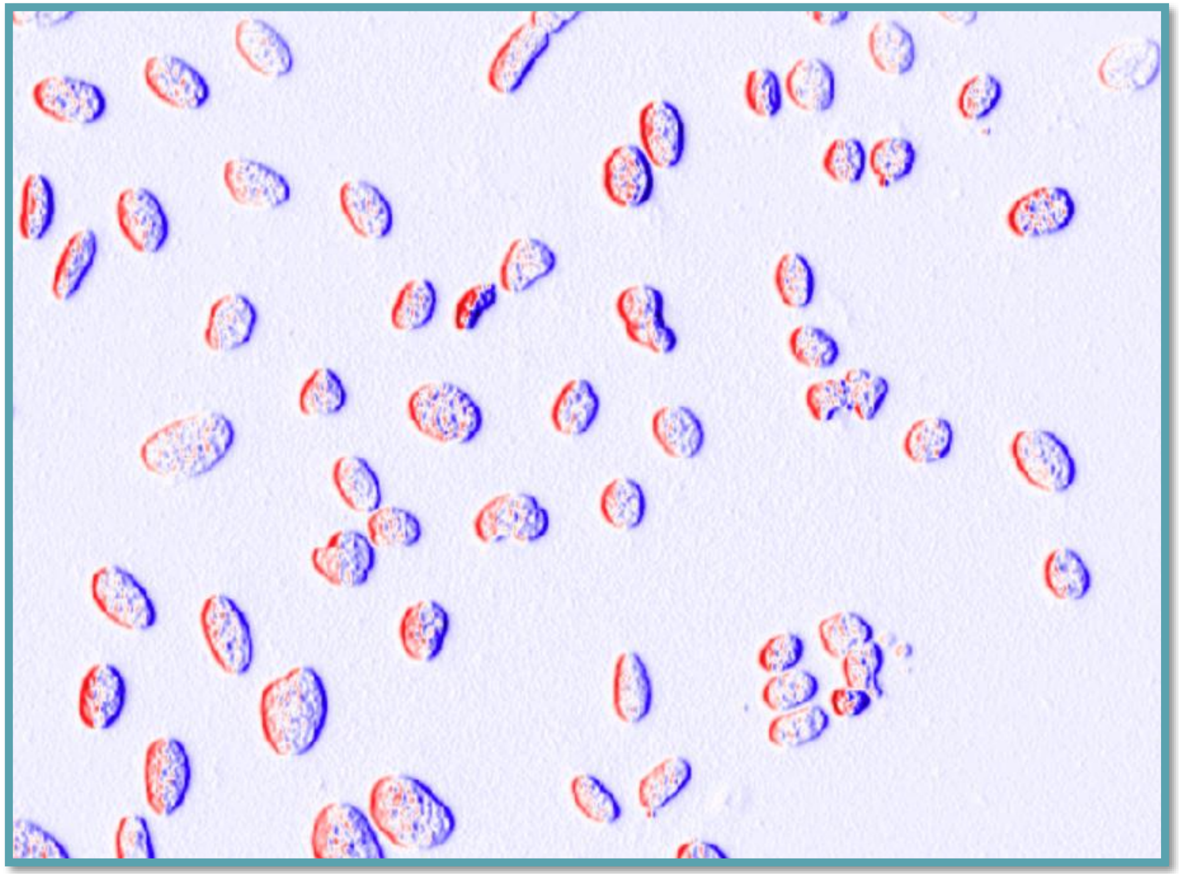
https://www.youtube.com/watch?v=IOHayh06LJ4


Mangocellular "Where?" pathway
Visual cortex
Pavocellular "What?" pathway

Convolution: Weighting an image by a pattern.

Horizontal Edges
3 x 3

1	0	-1
1	0	-1
1	0	-1



Convolution sums the element-wise product of an image window and filter

Image Window

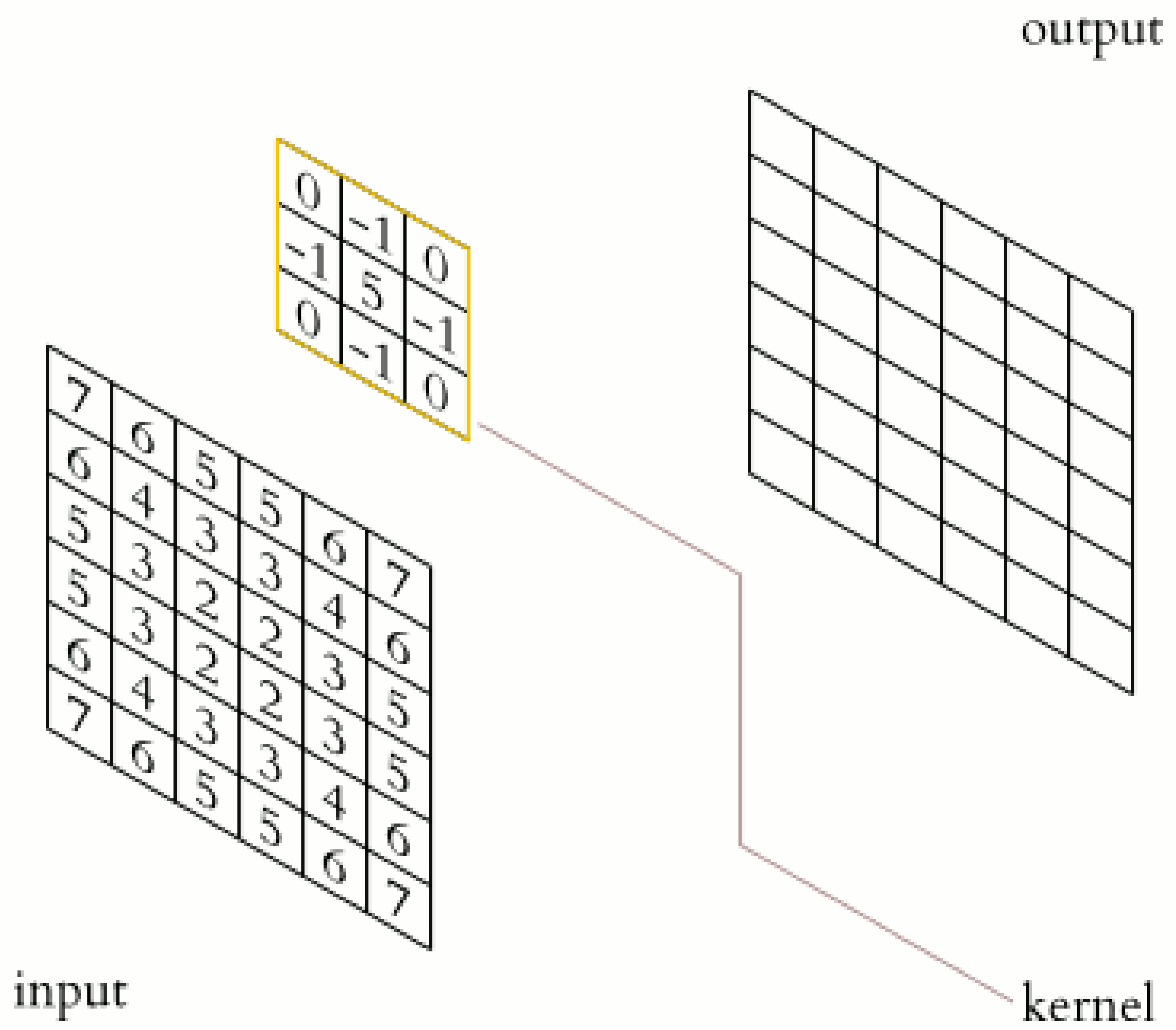
5	4	3
5	4	3
5	4	3

*

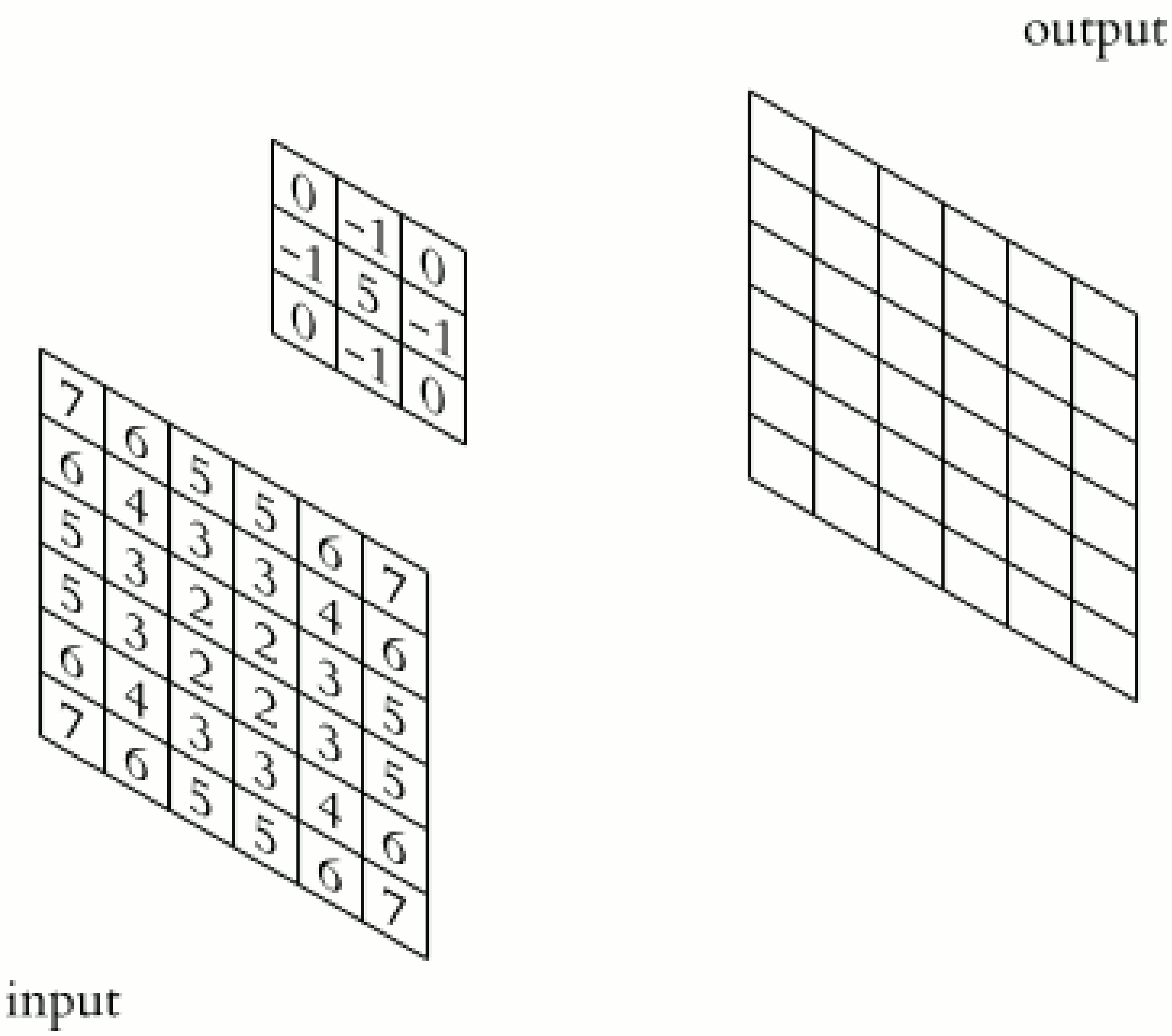
Filter

1	0	-1
1	0	-1
1	0	-1

A separate convolution is performed for each pixel.

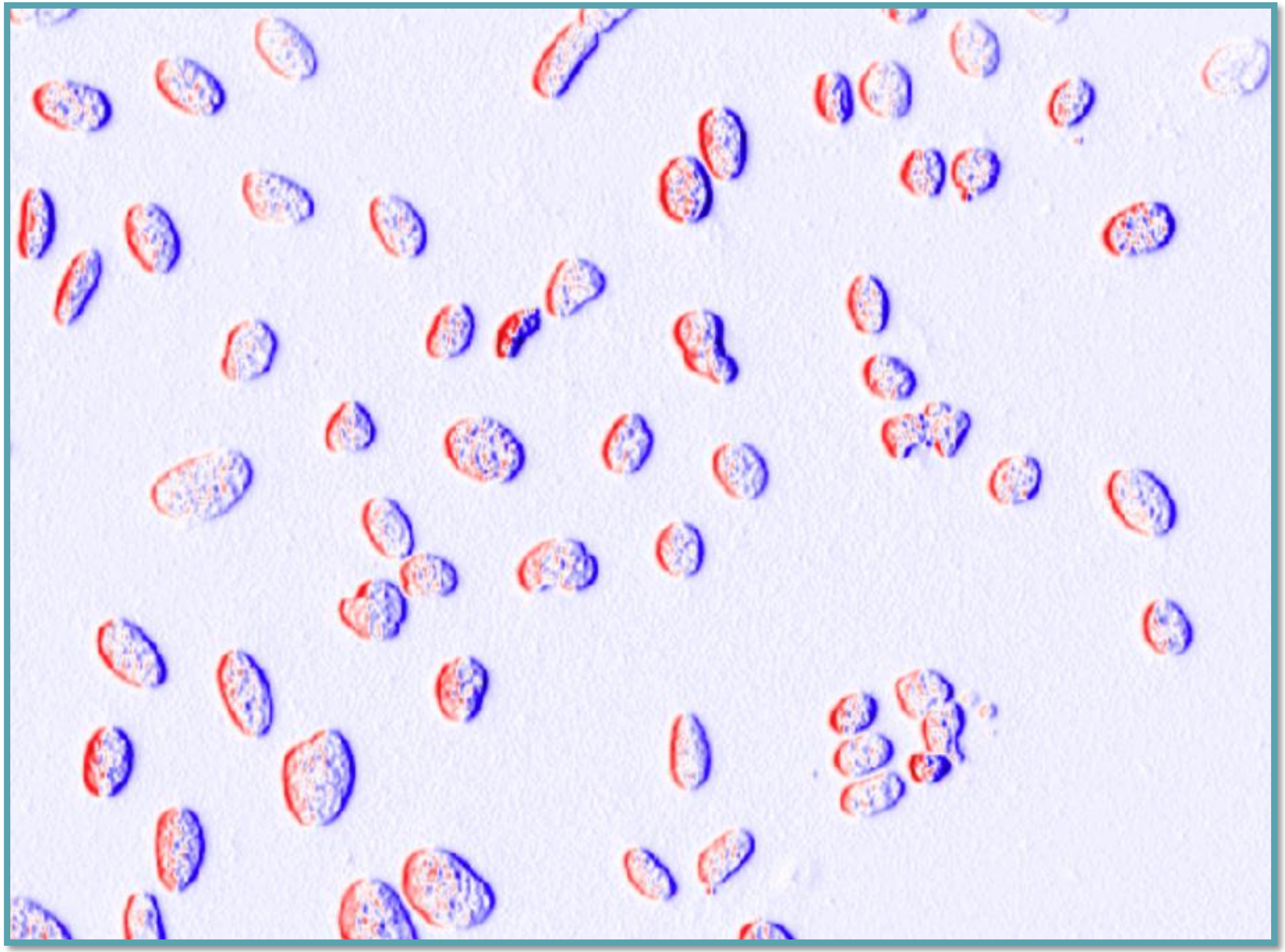


A separate convolution is performed for each pixel.



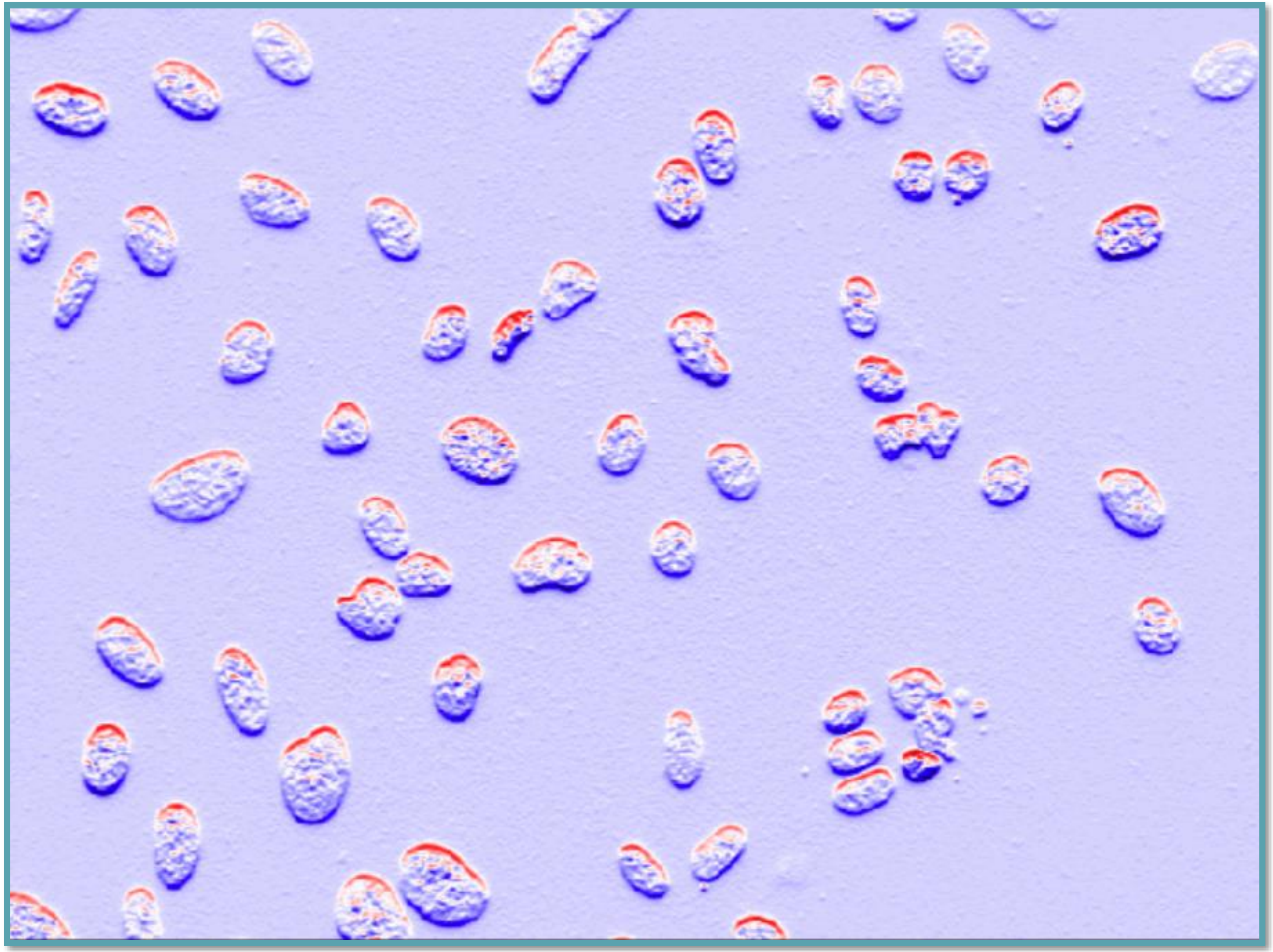
Results: Horizontal Edges

1	0	-1
1	0	-1
1	0	-1



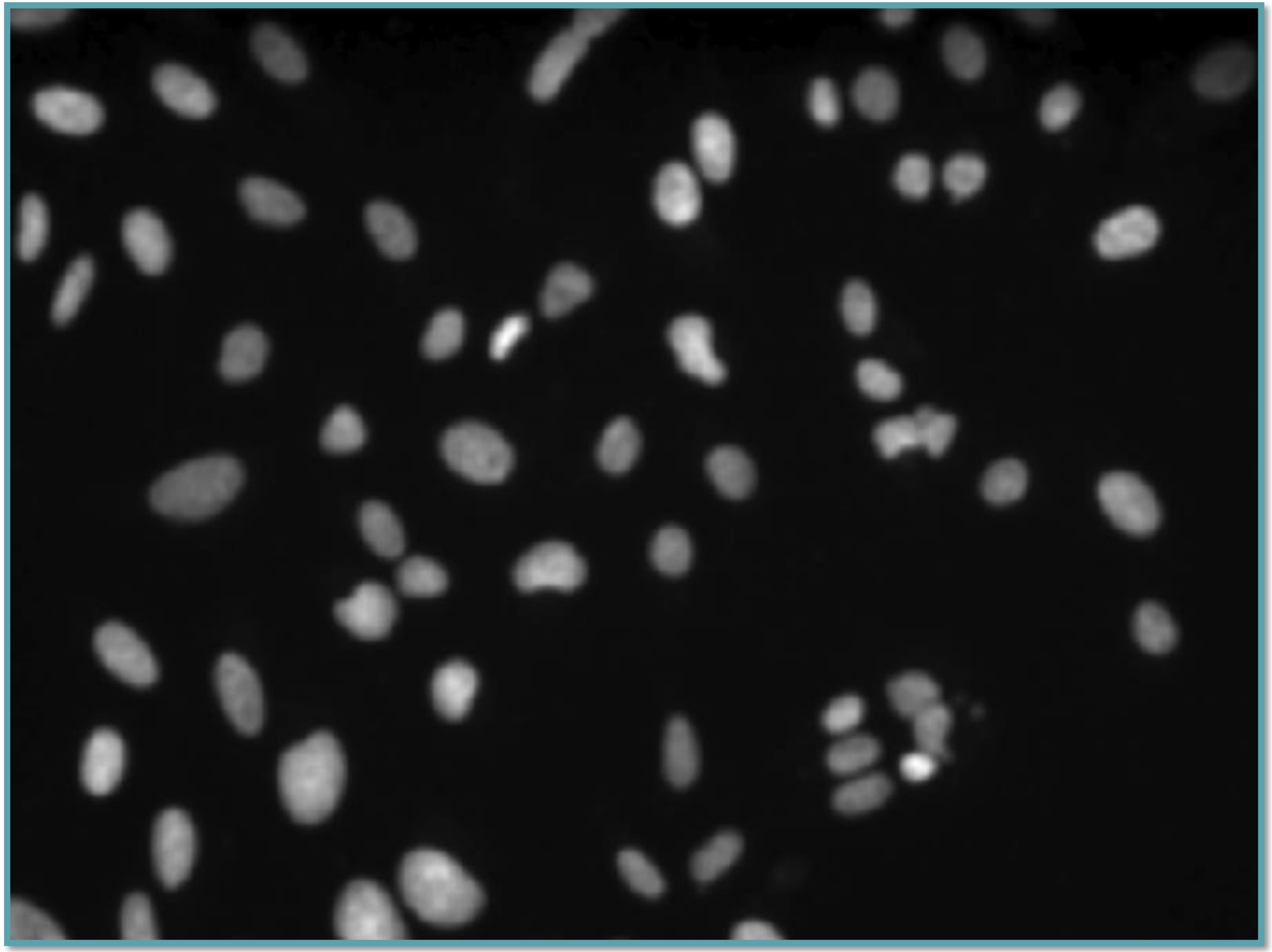
Results: Vertical Edges

1	1	1
0	0	0
-1	-1	-1



Results: Mean Filter

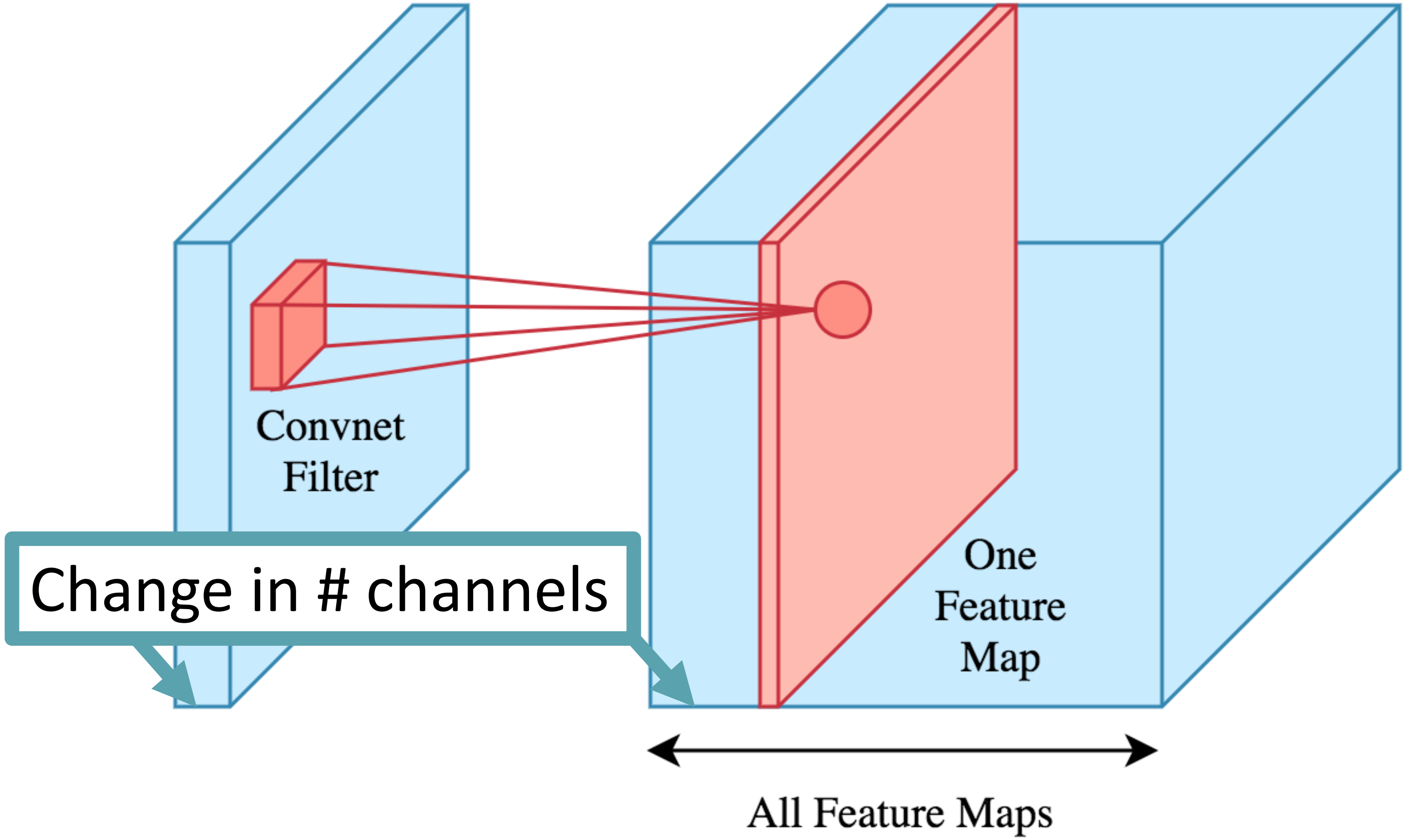
1	1	1
1	1	1
1	1	1



Convolve image with multiple filters

Input

Convolve image with many filters



- Horiz. edges
- Vert. edges
- Diag. edges
- Flat areas
- Smooth areas
- High color
- …

Convolutional neural networks

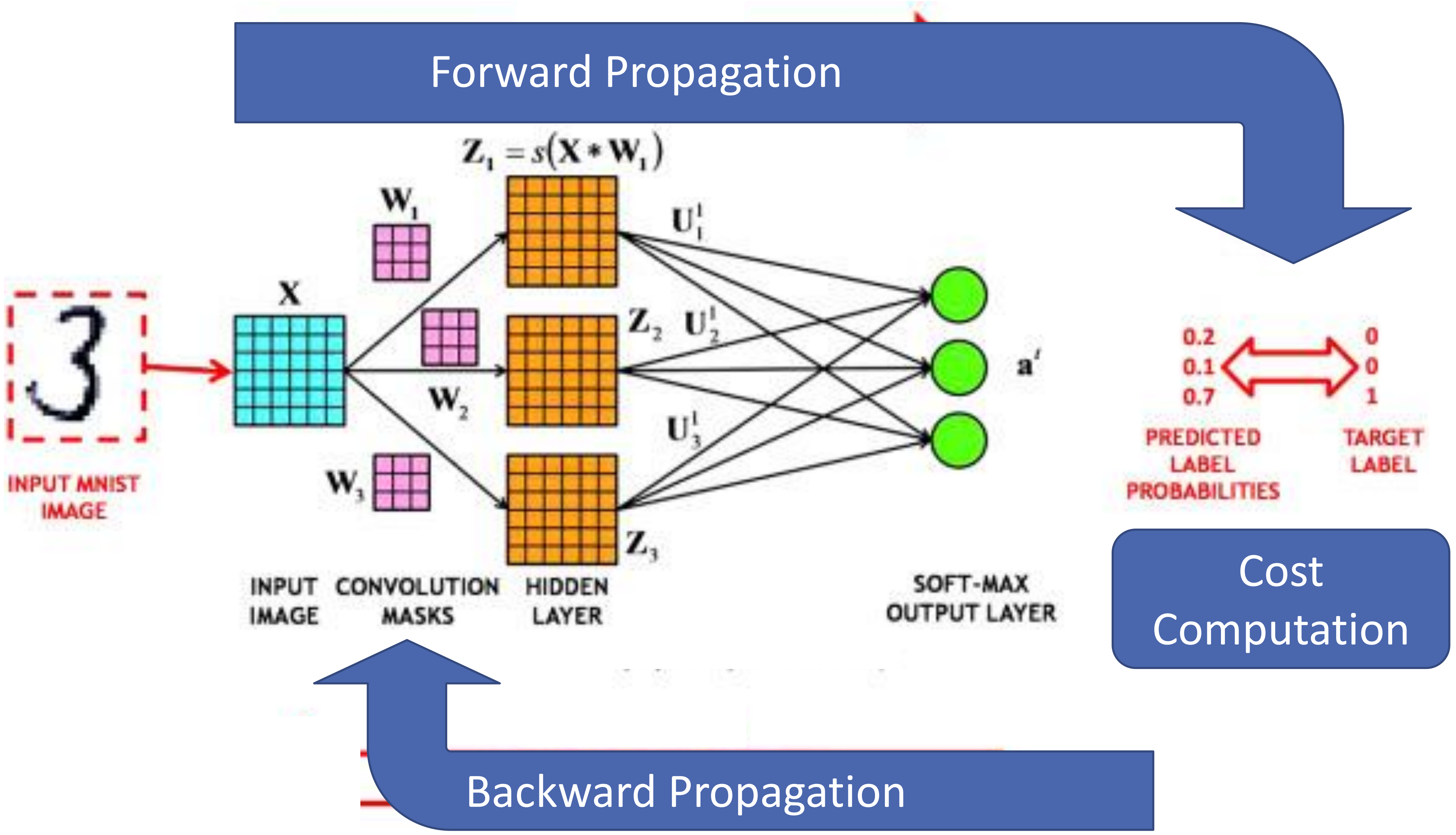
Don't impose filter weights… learn them!

Horizontal Edges
3 x 3

1	0	-1
1	0	-1
1	0	-1

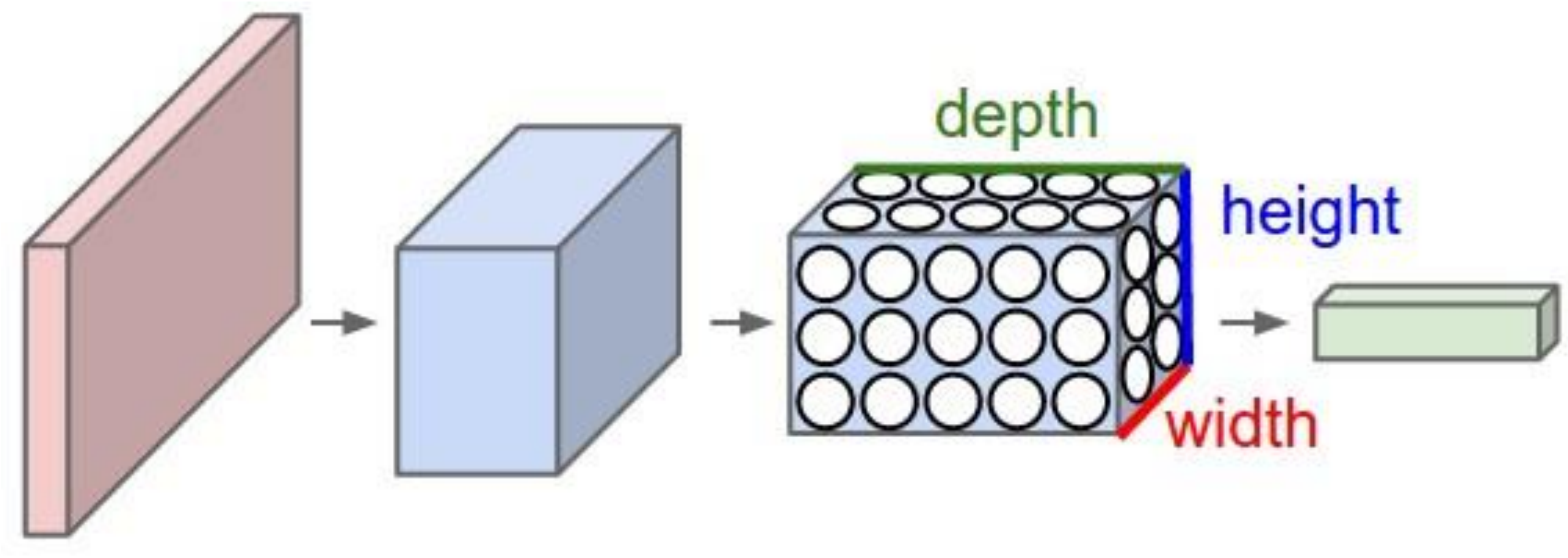
Conv2D Weights [0]
3 x 3

1.2	-0.3	-1.3
1.8	0.1	-1.6
1.1	0.2	-0.4

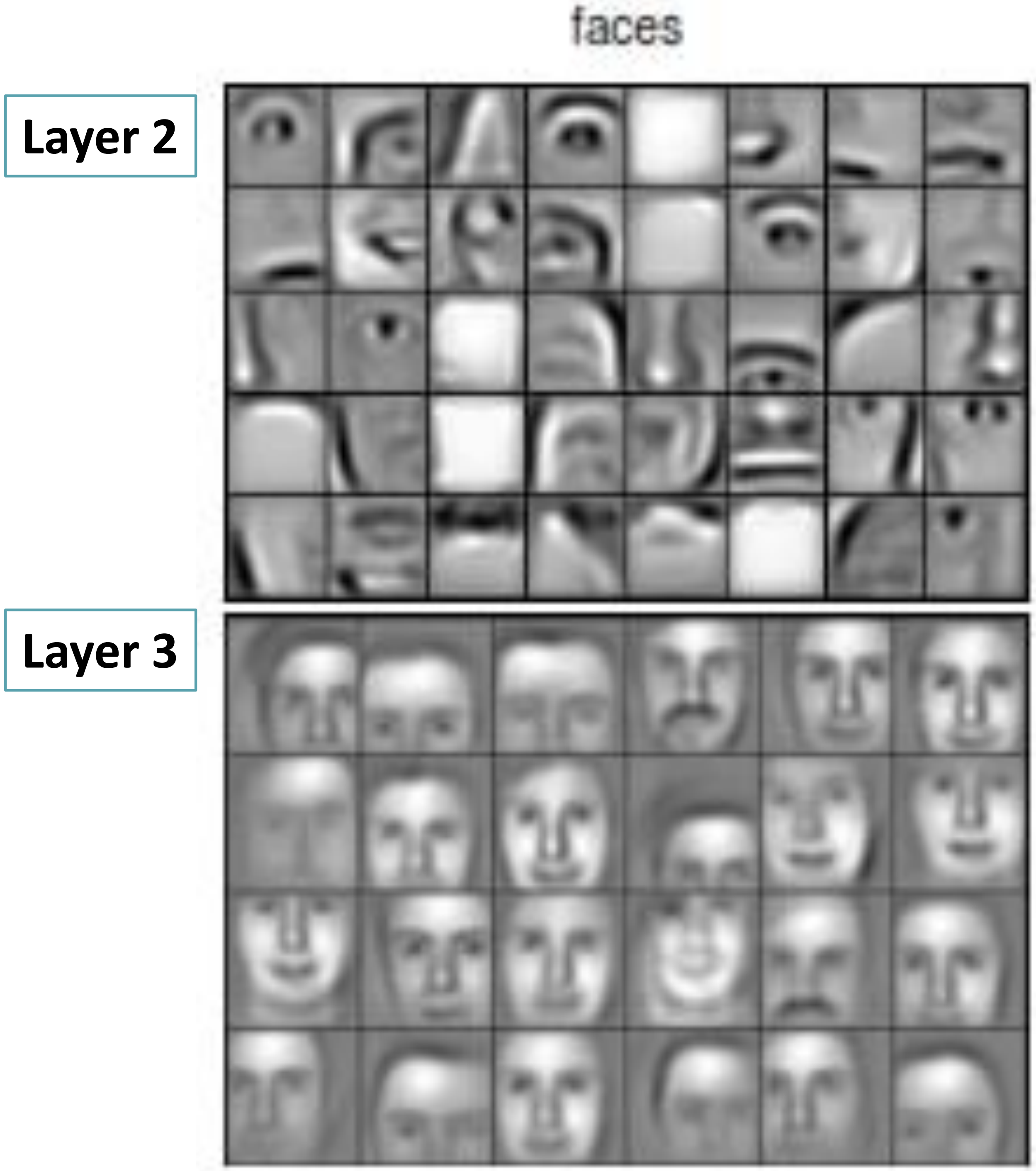


Forward Propagation
$Z_1 = s(X * W_1)$
W_1
U_1^1
X
Z_2
U_2^1
W_2
U_3^1
W_3
Z_3
a^l
3
INPUT MNIST IMAGE
INPUT IMAGE
CONVOLUTION MASKS
HIDDEN LAYER
SOFT-MAX OUTPUT LAYER
0.2
0.1
0.7
0
0
1
PREDICTED LABEL PROBABILITIES
TARGET LABEL
Cost Computation
Backward Propagation

Rule of thumb:
Layers get more compact but deeper.

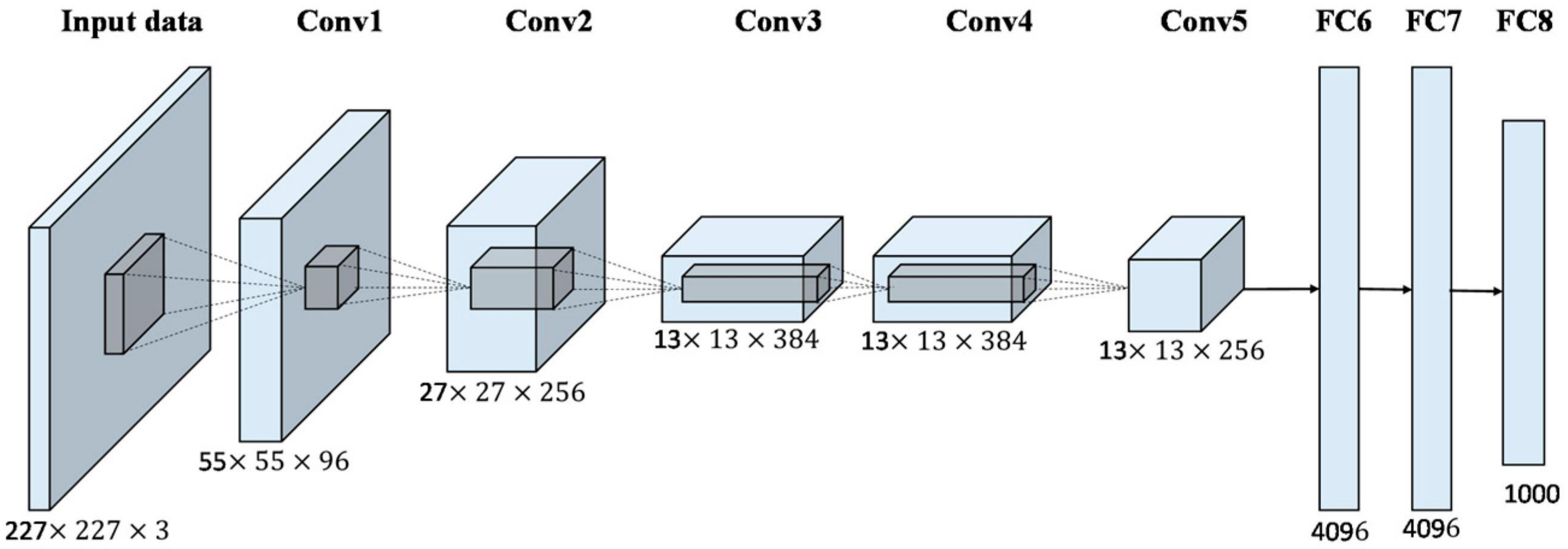


Rule of thumb: Filters become sensitive to higher level features in deeper layers



www.chioka.in

Rule of thumb: Use a pre-built architecture



https://medium.com/

Topics

The 2018 Data Science Bowl

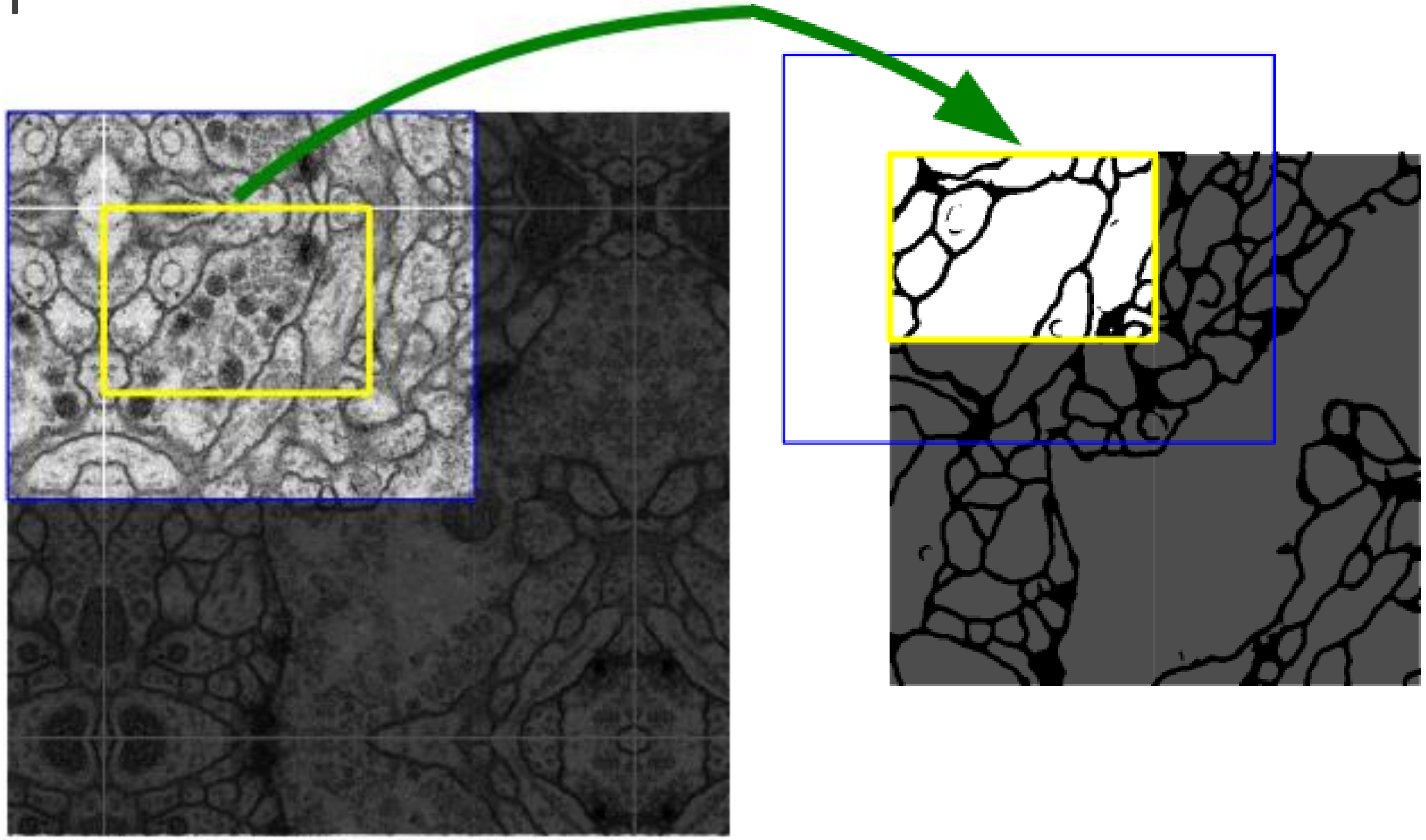
Puppy bowl

- Image processing basics
- The simple approach

Superbowl

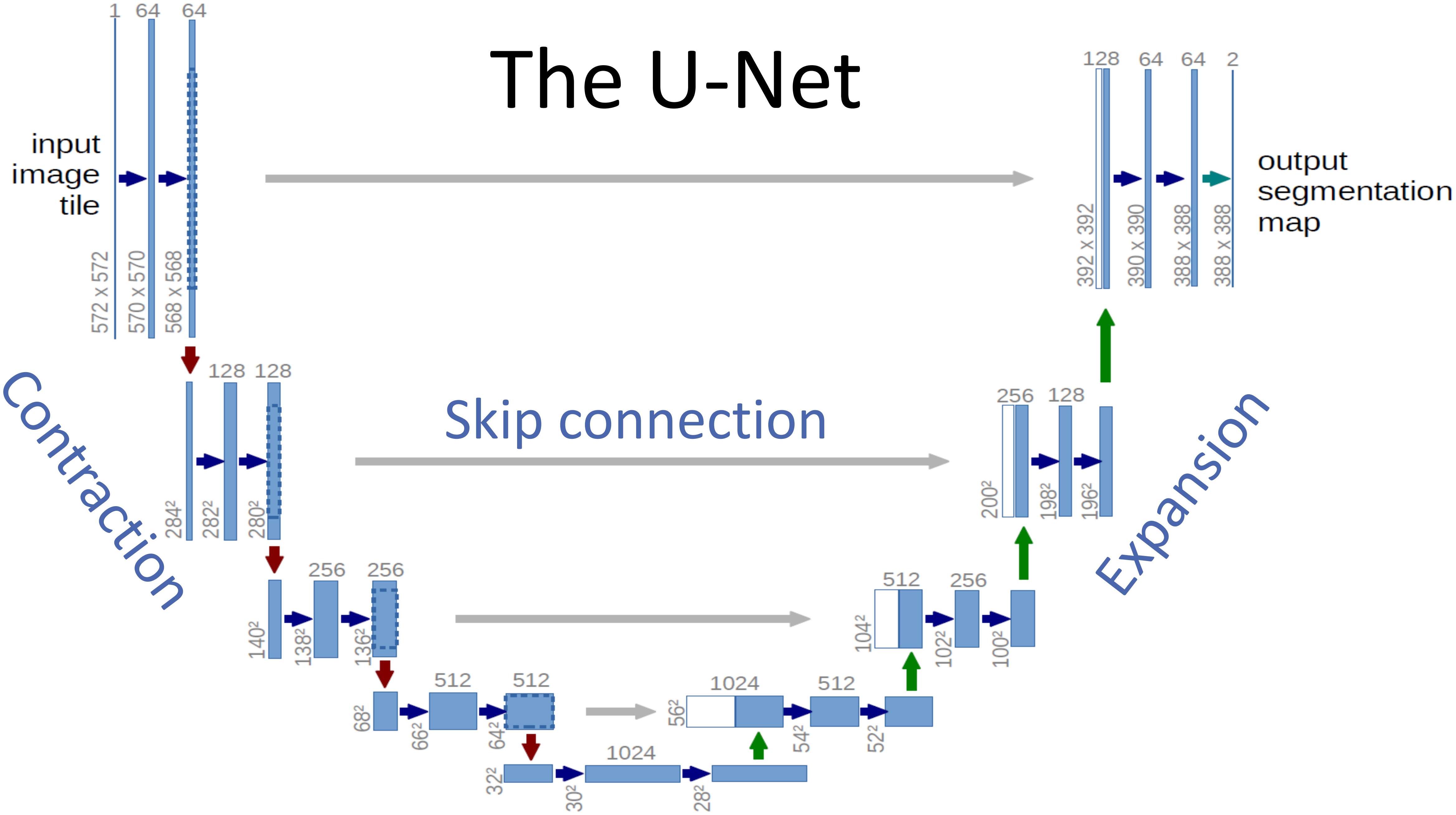
- Feature detection basics
- **The U-Net approach**

Performance



Ronneberger et al., 2015

The U-Net



Ronneberger et al., 2015

Layer Types

Convolutions

- 3 x 3 Filter Size
- “Valid” padding

Max Pooling

Up-convolutions

Skip Connections

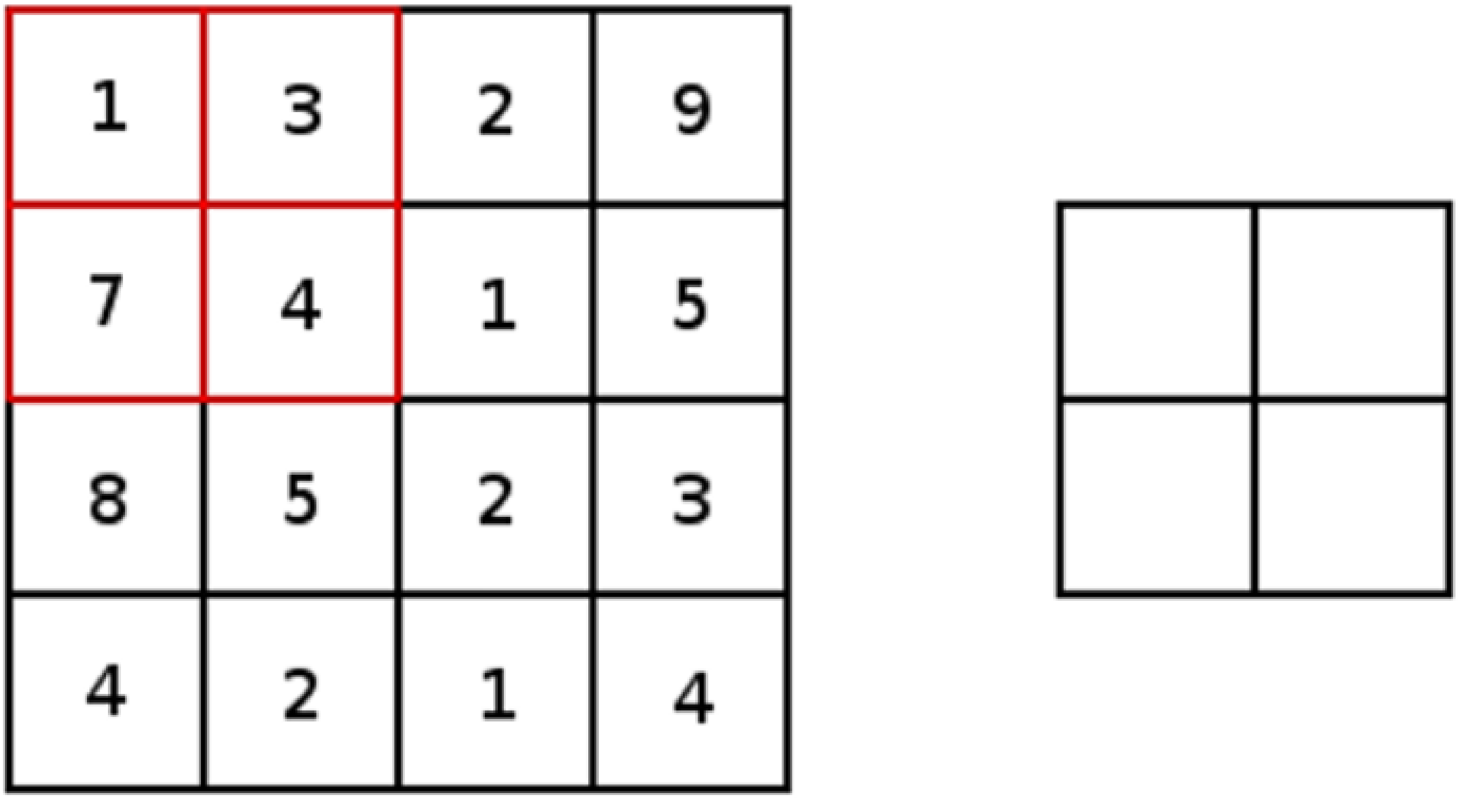
Layer Types

Convolutions
- 3 x 3 Filter Size
- "Valid" padding

Max Pooling

Up-convolutions

Skip Connections



Layer Types

Convolutions
- 3 x 3 Filter Size
- “Valid” padding

Max Pooling

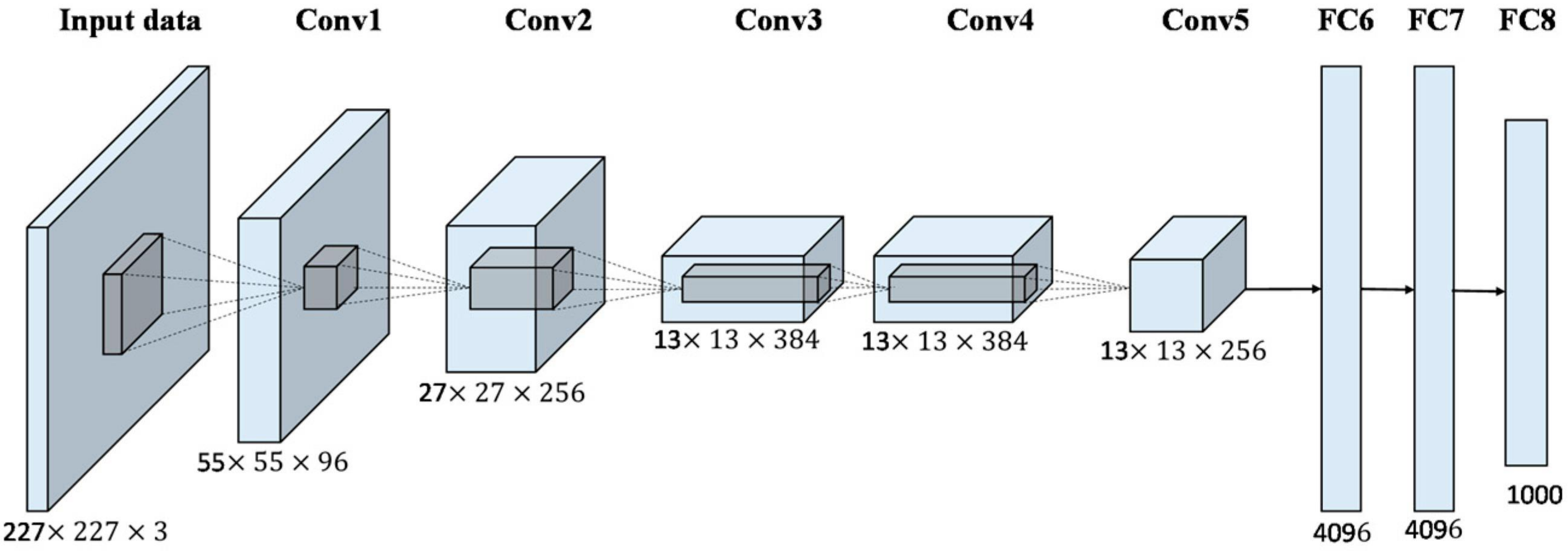
Up-convolutions

Skip Connections

1	3	2	9
7	4	1	5
8	5	2	3
4	2	1	4

7	9
8	

Convolutions change depth, Pooling changes height / width



https://medium.com/

Layer Types

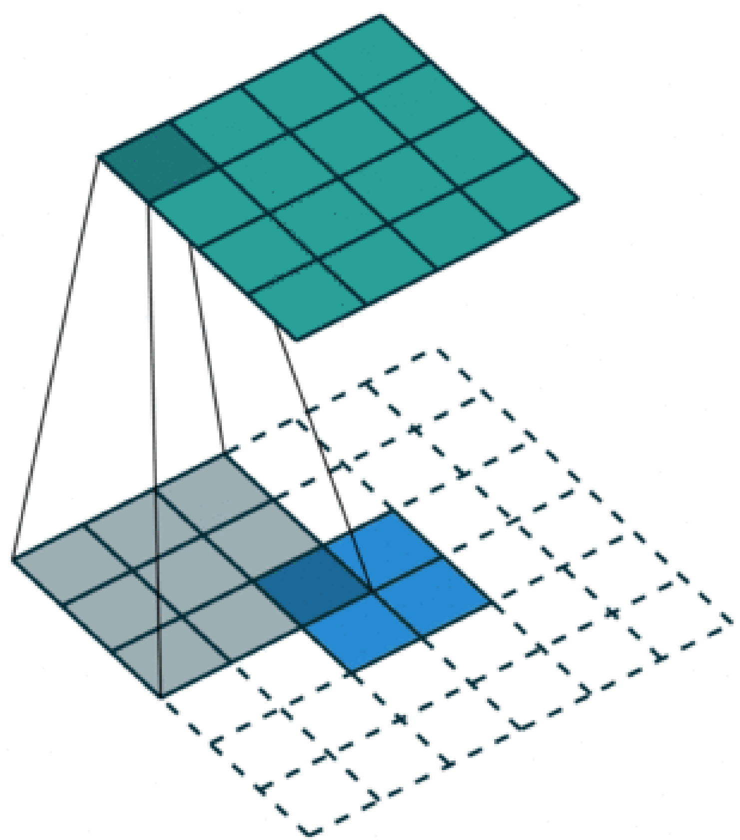
Convolutions

- 3 x 3 Filter Size
- “Valid” padding

Max Pooling

Up-convolutions

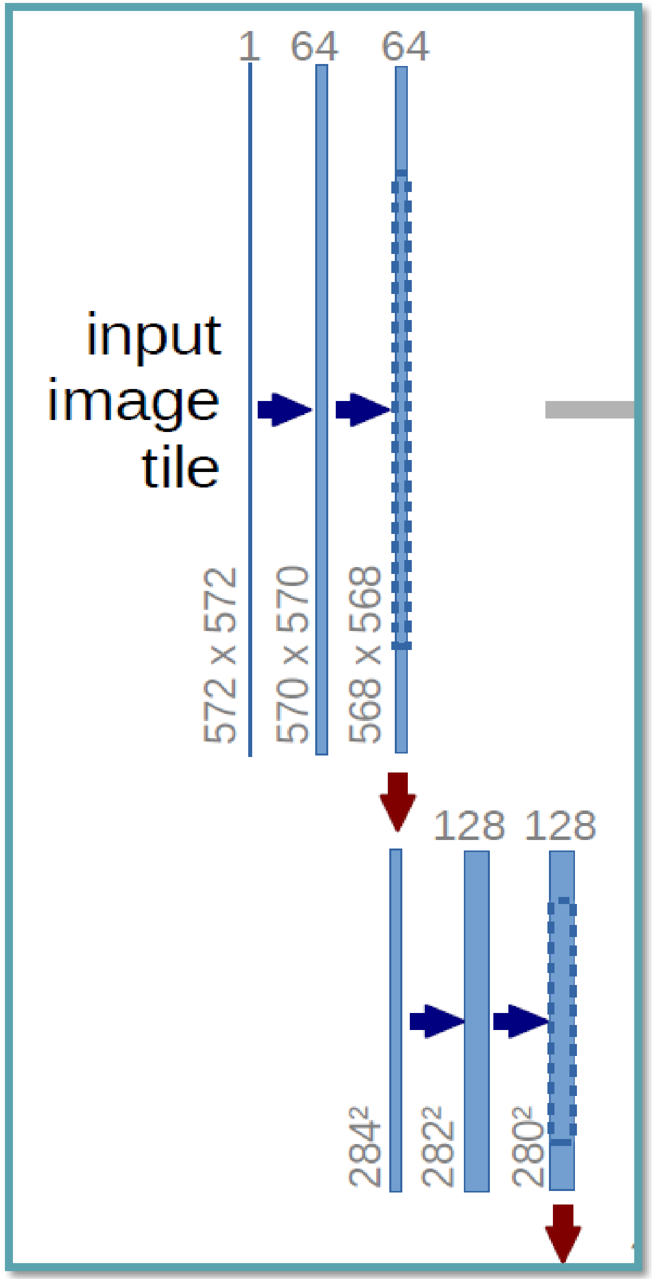
Skip Connections



https://github.com/vdumoulin/conv_arithmetic

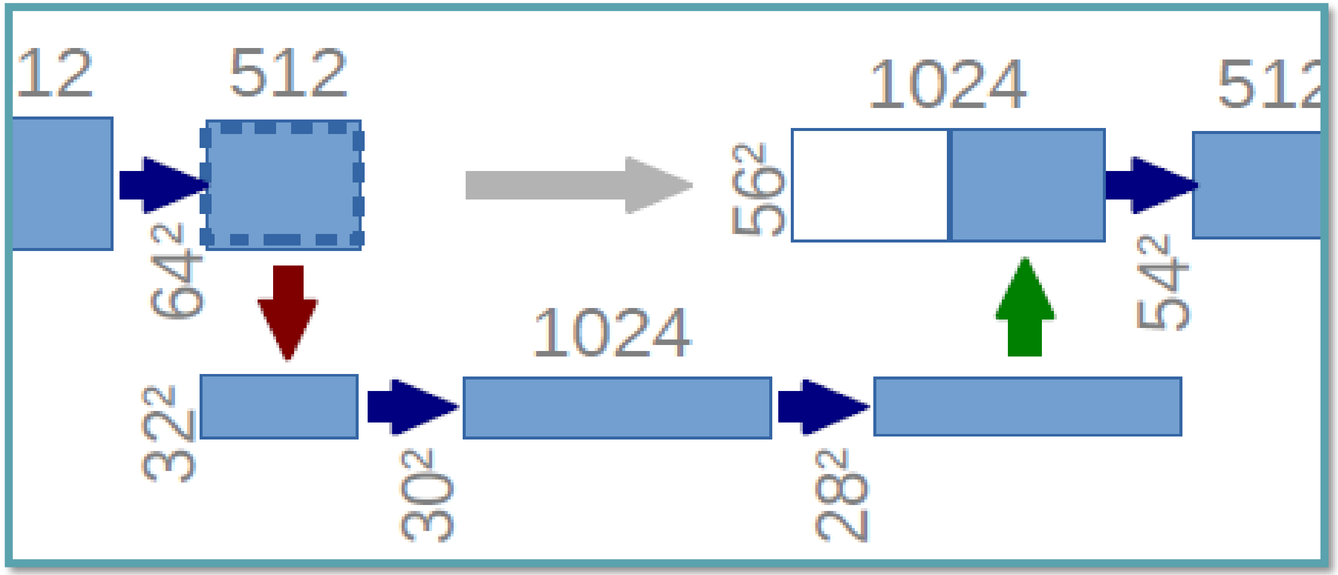
Contraction Path

Layer	Area	Depth
Input	572^2	1
Conv1	570^2	64
Conv2	568^2	64
MaxPool1	284^2	64
Conv3	282^2	128
Conv4	280^2	128
MaxPool2	140^2	128



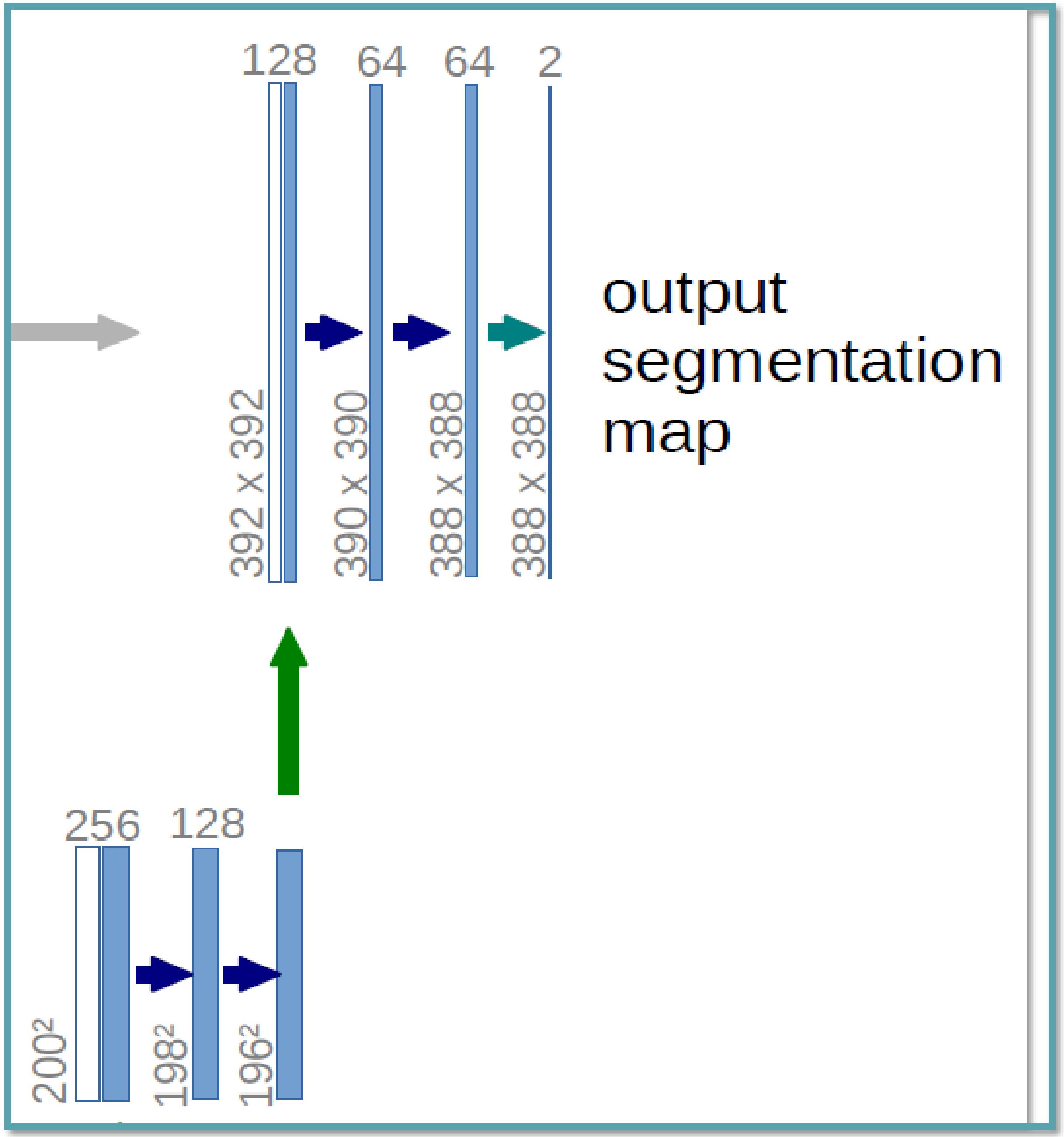
Basin

Layer	Area	Depth
MaxPool4	32^2	512
Conv9	30^2	1024
Conv10	28^2	1024
UpConv1	56^2	512



Expansion Path

Layer	Area	Depth
UpConv3	200^2	256
Conv15	198^2	128
Conv16	196^2	128
UpConv4	392^2	128
Conv17	390^2	64
Conv18	388^2	64
Output	388^2	2



Skip Connections

Crops and copies information from earlier in the network to deeper layers

Allows final resolution to remain high

kaggle Search kaggle Competitions Datasets Kernels Discussion Learn •••

Kjetil Åmdal-Sævik

Keras U-Net starter - LB 0.277

747 voters

last run 3 months ago · Python notebook · 63592 views
using data from 2018 Data Science Bowl · Public

Notebook Code Data (1) Output (2) Comments (126) Log Versions (8) Forks (465) Fork Notebook

You are currently viewing an old version of this script (7/8). View the latest version.

Tags tutorial

Notebook

```python
# Build U-Net model
inputs = Input((IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS))
s = Lambda(lambda x: x / 255) (inputs)

c1 = Conv2D(8, (3, 3), activation='relu', padding='same') (s)
c1 = Conv2D(8, (3, 3), activation='relu', padding='same') (c1)
p1 = MaxPooling2D((2, 2)) (c1)

c2 = Conv2D(16, (3, 3), activation='relu', padding='same') (p1)
c2 = Conv2D(16, (3, 3), activation='relu', padding='same') (c2)
p2 = MaxPooling2D((2, 2)) (c2)
```

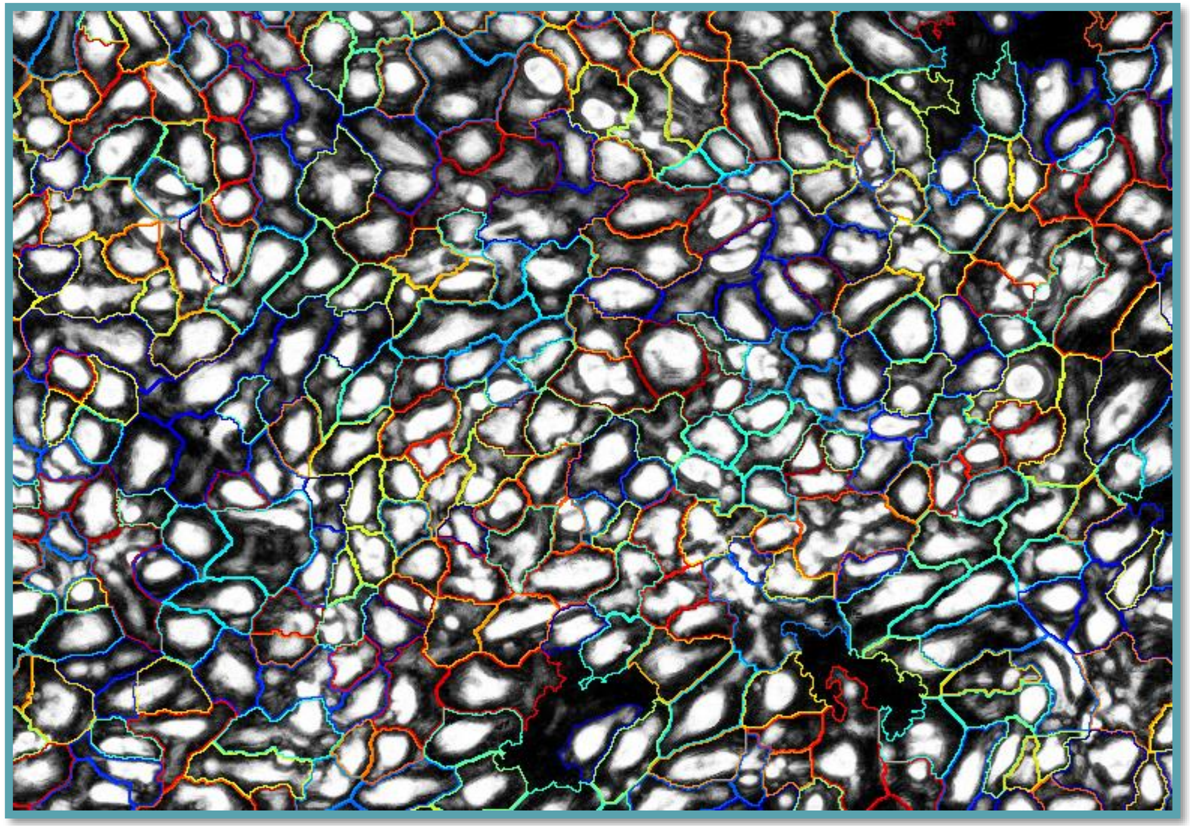
Topics

The 2018 Data Science Bowl

Puppy bowl
- Image processing basics
- The simple approach

Superbowl
- Feature detection basics
- The U-Net approach

Performance



Model performance (0.0 – 0.6)

Approach	Mean Average Precision
1. Vanilla image processing	0.20 – 0.28
2. Vanilla U-Net	0.25 – 0.42
3. Vanilla Mask R-CNN	0.37 – 0.50
4. U-Net + Deep Watershed	0.45 – 0.55
5. Winning model	~ 0.65

Insight 1:
Data augmentation is critical.

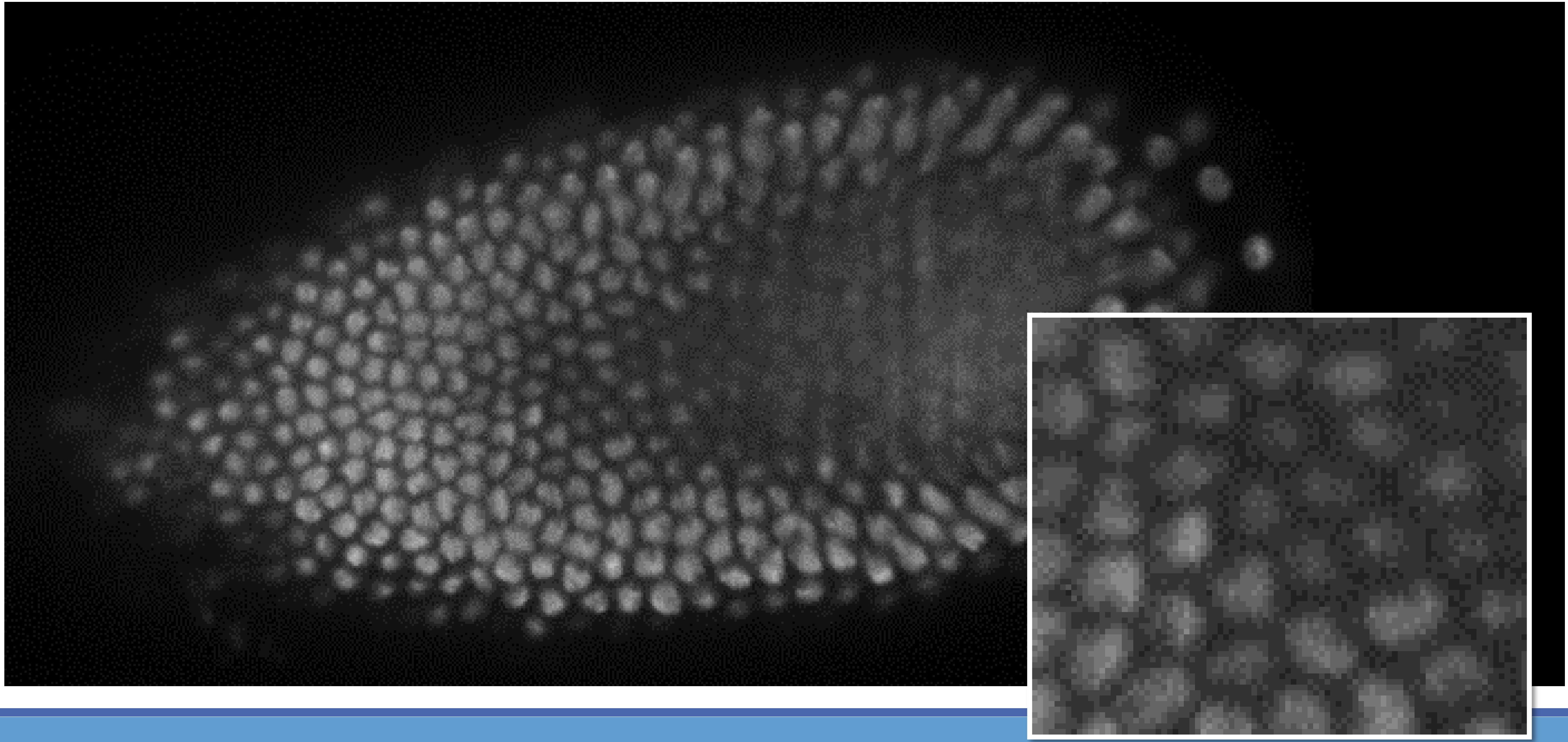
1. Add gaussian noise
2. Color to gray
3. Contrast and brightness
4. Random crop
5. Mosaics
6. Random rotate

Insight 2:
Neural networks don’t do it all.

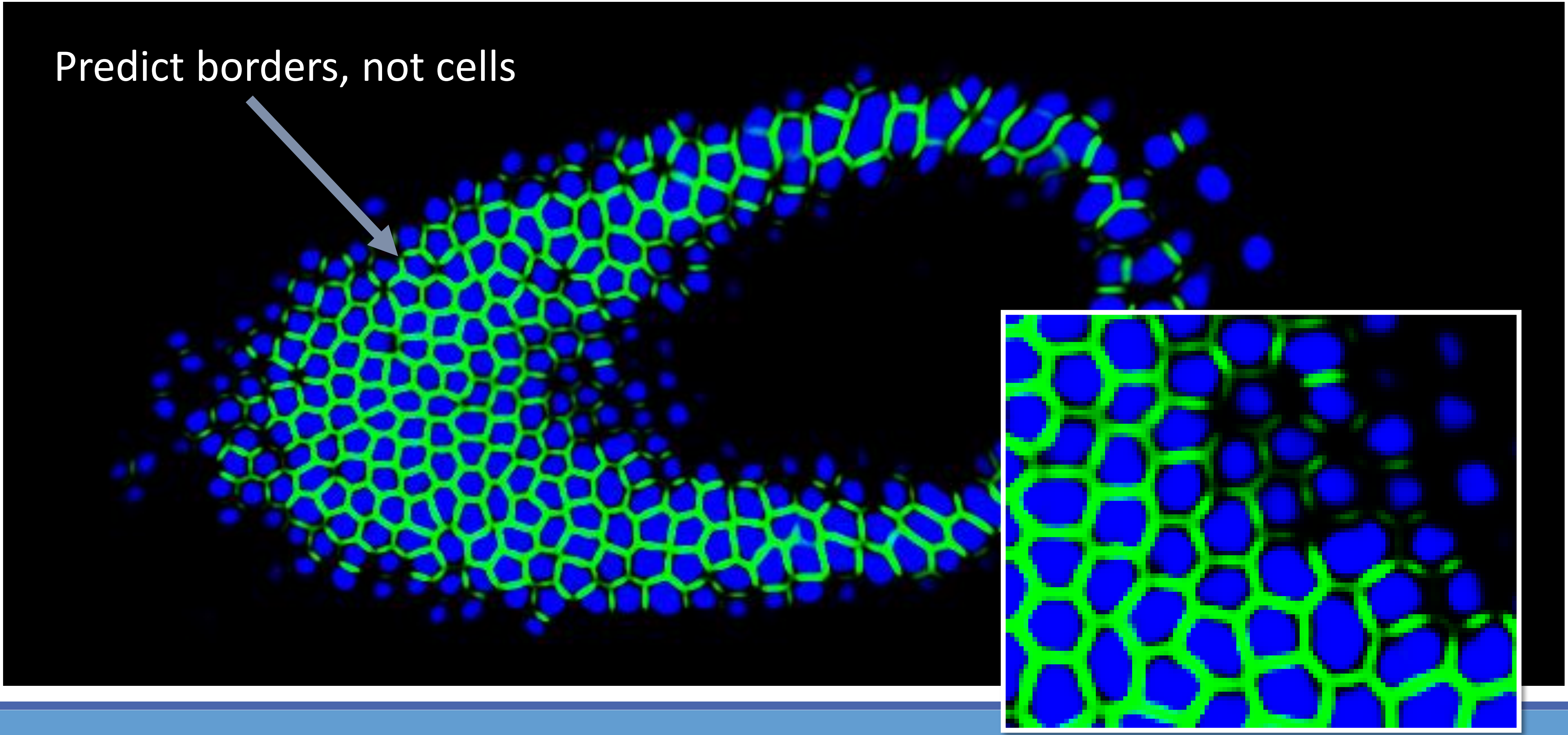
Output of models was treated as “candidate nuclei”… still need to post-process data!

- e.g., Use morphological features such as solidity, circularity, convexity, area, neighbors median, count, etc. to calculate “predicted IOU” for each mask.

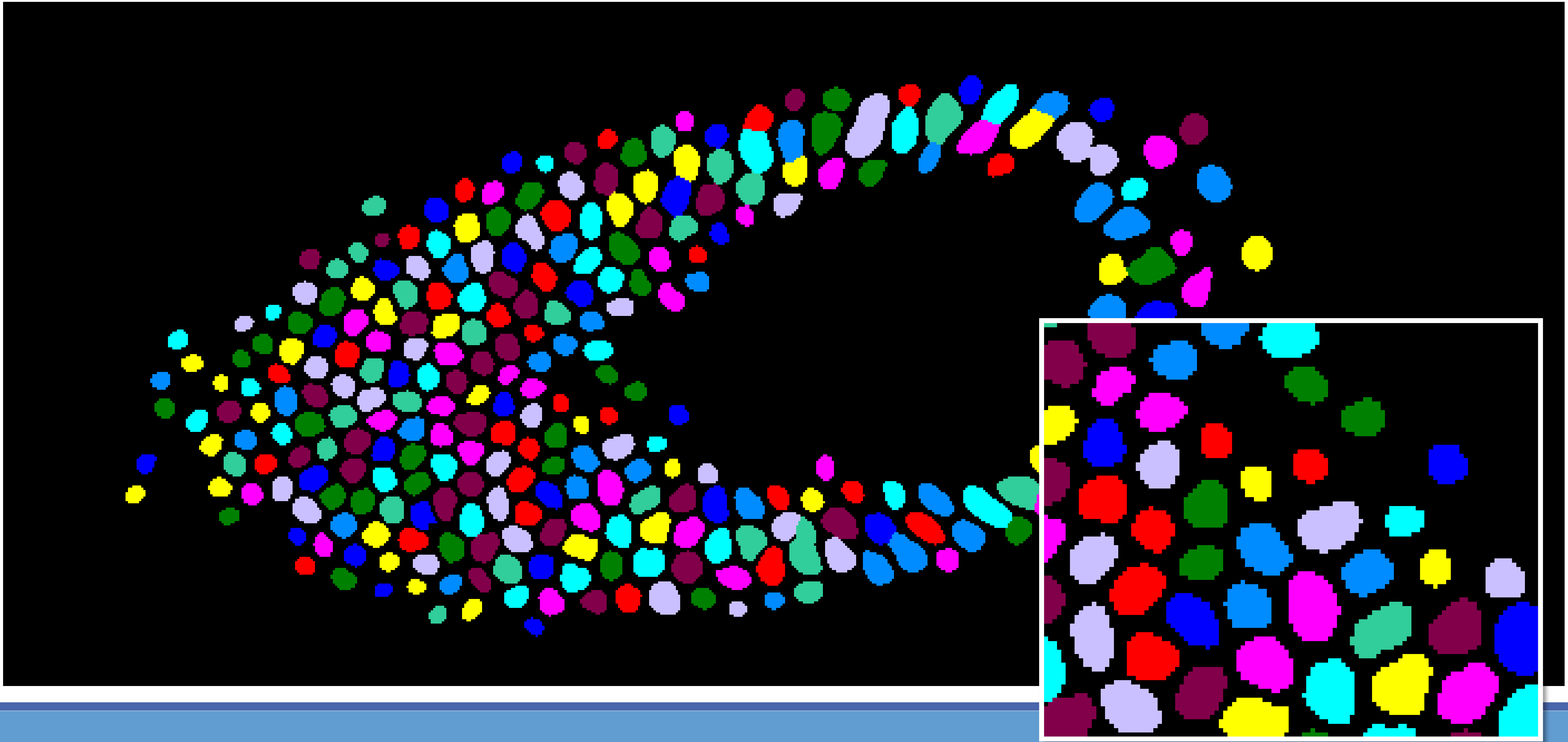
Insight 3: Pick the right target…



Insight 3: Pick the right target…



Insight 3: Pick the right target…



For further reference…

1. Coursera's Deep learning specialization
2. "U-Net: Convolutional Networks for Biomedical Image Segmentation" by Ronneberger, Fischer and Brox
3. Kaggle's Data Science Bowl 2018
   1. 1st Place Solution from *ods.ai*
   2. 4th Place U-Net + Watershed from *Nuclear Vision*
   3. 11th Place Open Solution from Zheng Li