

# How to Get Started in AI

Brian Anderson, MS

# For running on your own

- Go to google colab in a chrome browser
  - <https://colab.research.google.com/notebooks/intro.ipynb>
- Click 'File' -> 'open notebook'
  - Click 'GitHub' tab and search brianmanderson
- Select the Repository Imaging\_Physics\_Workshop\_1\_28\_20
  - Click 'Click\_Me.ipynb'
- Follow instructions to change Runtime to GPU
- All packages at [www.github.com/brianmanderson](http://www.github.com/brianmanderson)

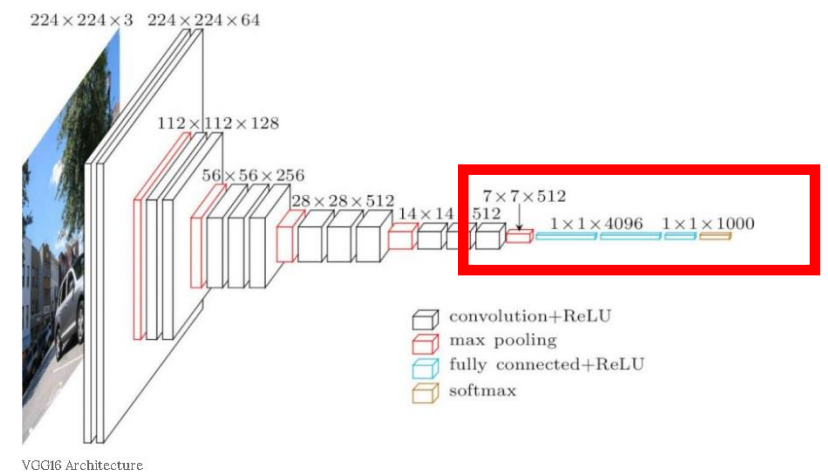
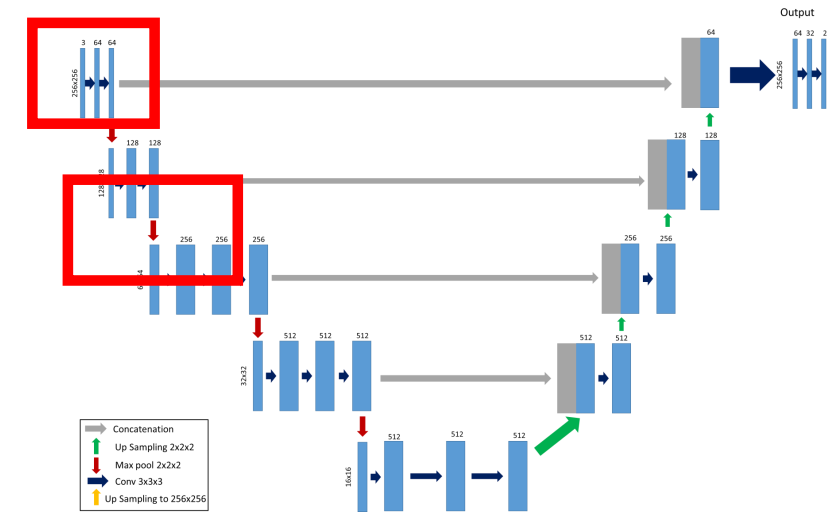
# Overview

- Machine\Deep learning in imaging
  - Convolutions!
    - Difference between deep and machine learning
- Creating data from medical images (provided)
- Things I wish I'd known

# Imaging (Convolution Nets)

Machine\Deep Learning Style

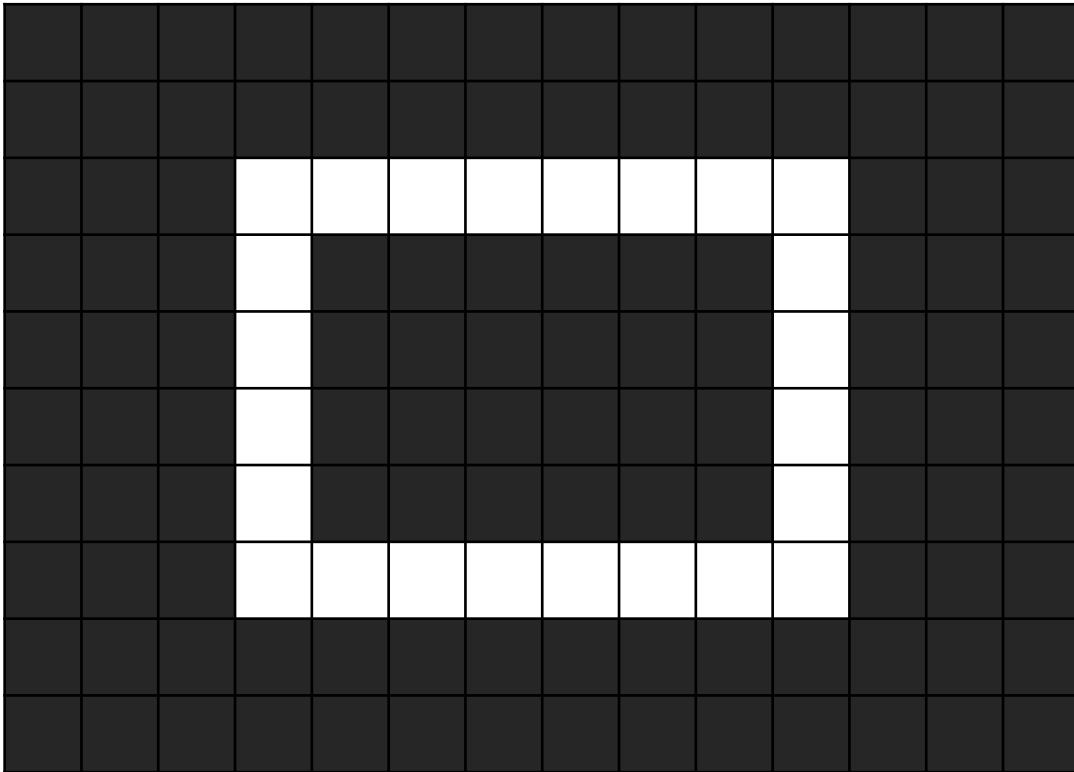
# 'Main parts' of convolution network



<https://neurohive.io/en/popular-networks/vgg16/>

# What is a convolution?

This is a box!



Two horizontal and two parallel lines

Convolution

Activation

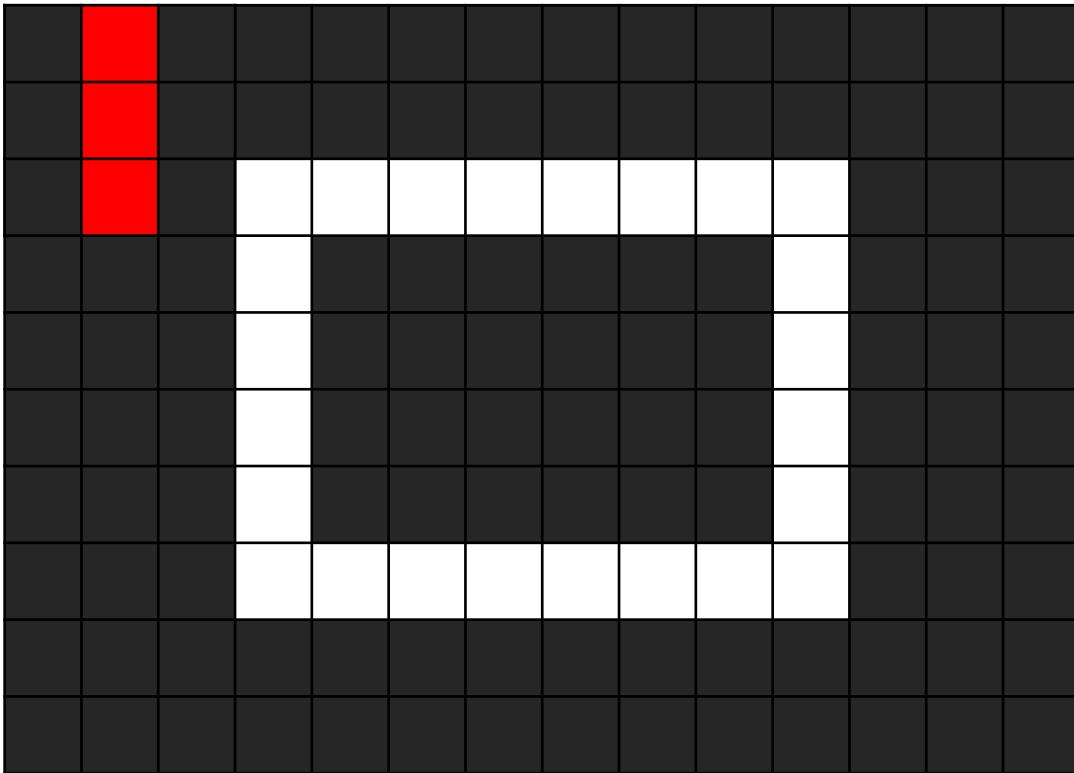
Max Pooling

Fully Connected

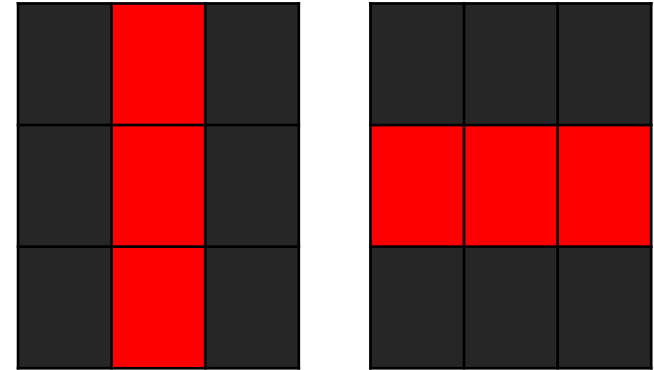


# What is a convolution?

This is a box!



Kernels



Convolution

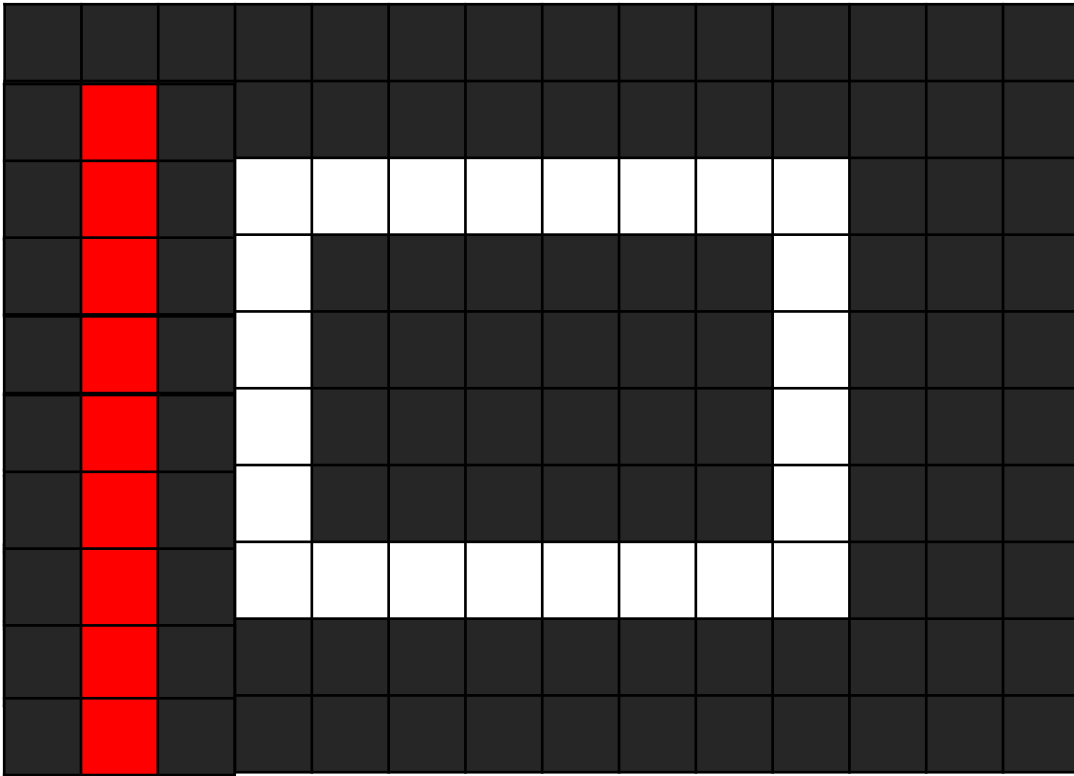
Activation

Max Pooling

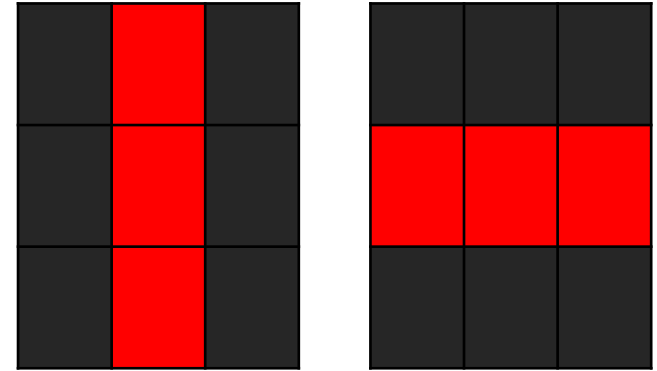
Fully Connected

# What is a convolution?

This is a box!



Kernels



Convolution

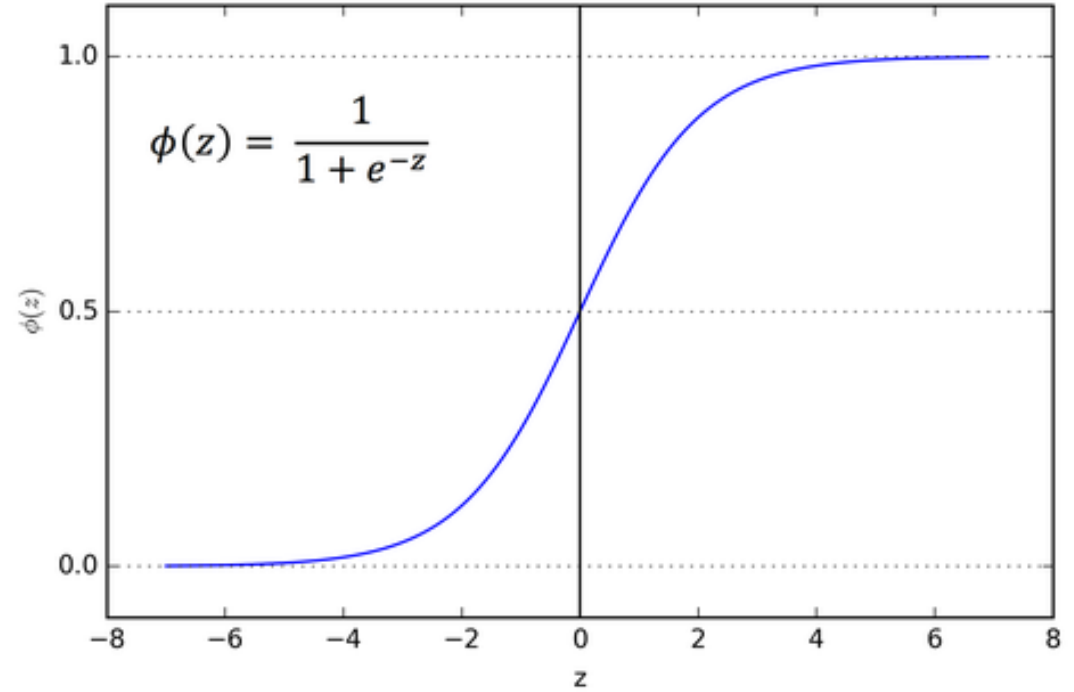
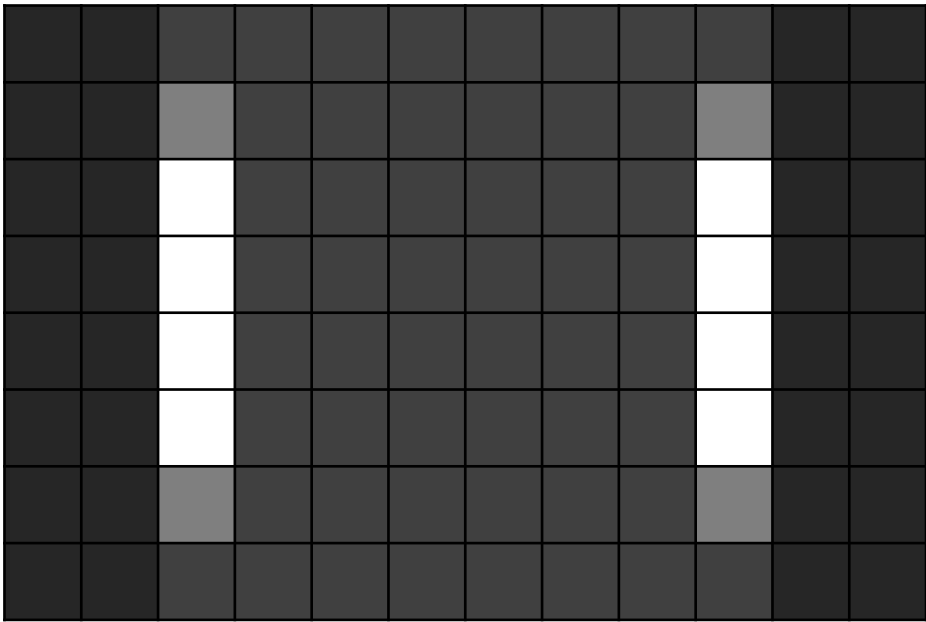
Activation

Max Pooling

Fully Connected



# Activation



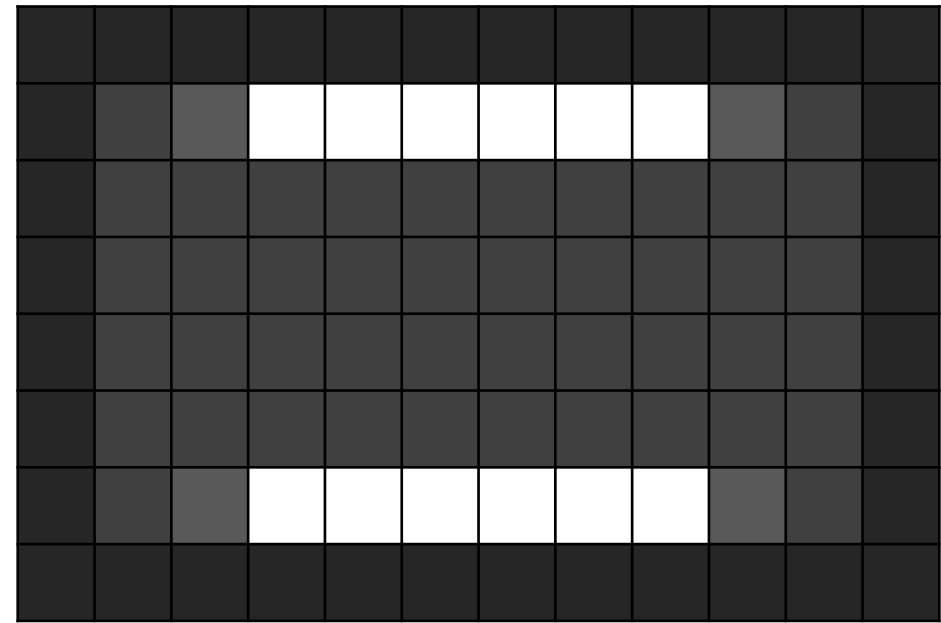
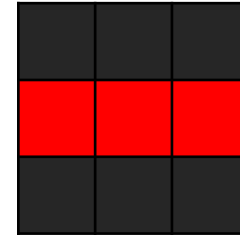
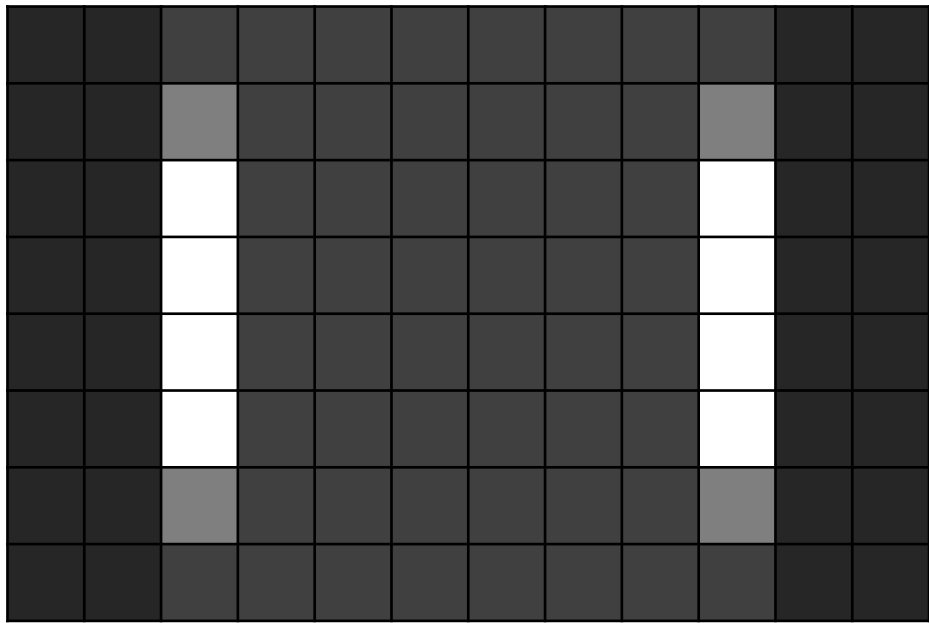
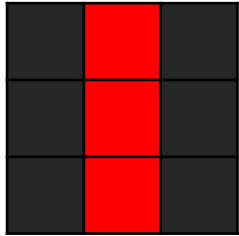
Convolution

Activation

Max Pooling

Fully Connected

# Horizontal and Vertical...



Convolution

Activation

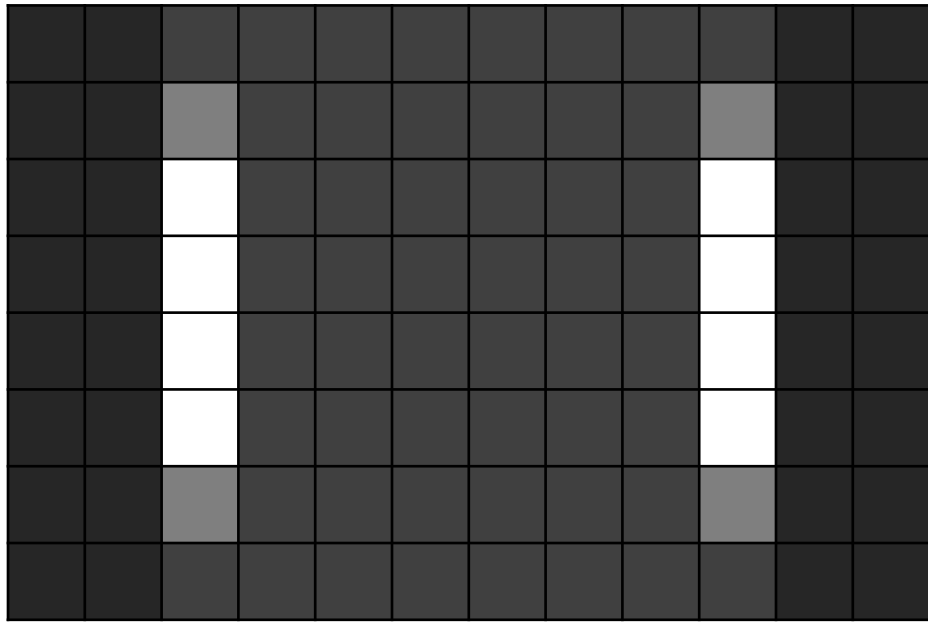
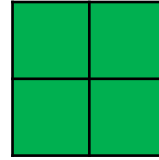
Max Pooling

Fully Connected



# Max Pooling

Max value in a 2x2 region



Convolution

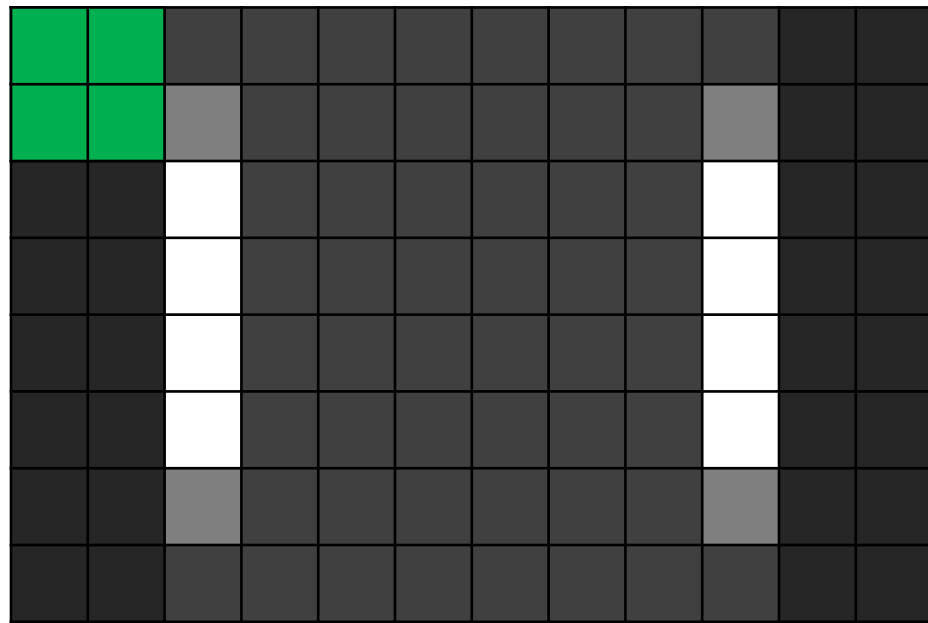
Activation

Max Pooling

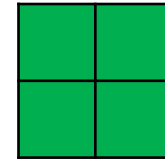
Fully Connected



# Max Pooling



Max value in a 2x2 region



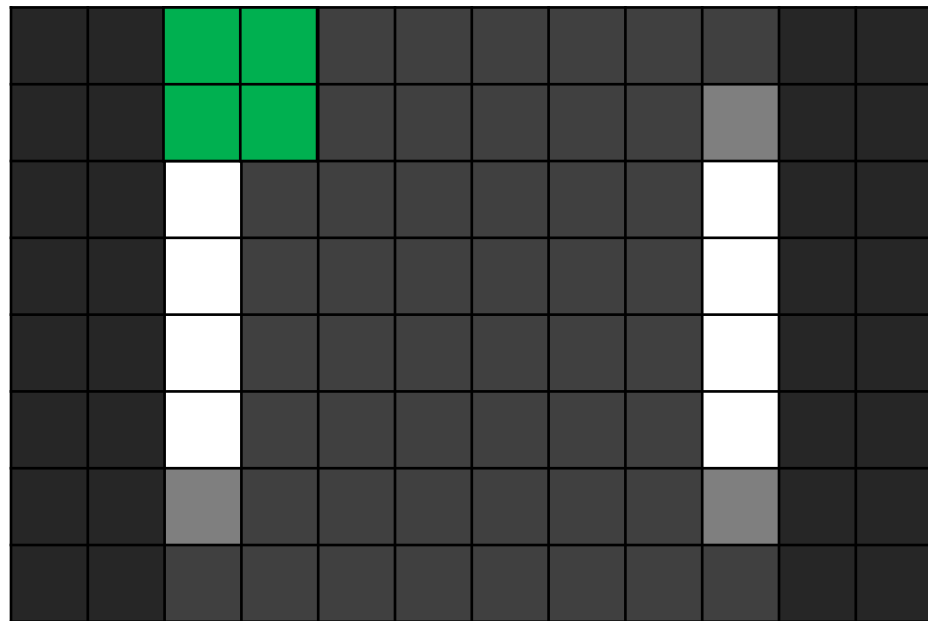
Convolution

Activation

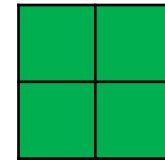
Max Pooling

Fully Connected

# Max Pooling



Max value in a 2x2 region



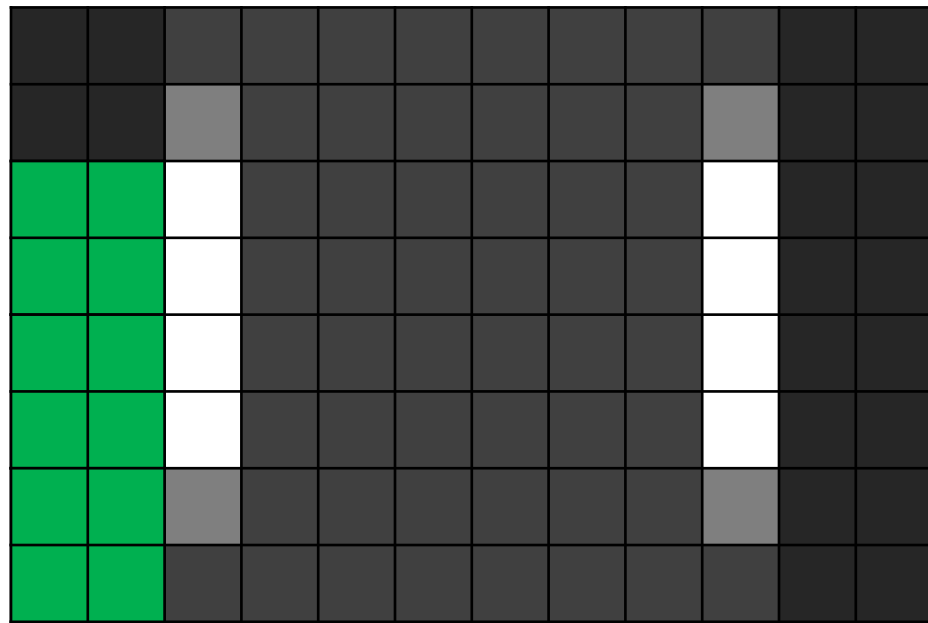
Convolution

Activation

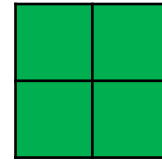
Max Pooling

Fully Connected

# Max Pooling



Max value in a 2x2 region



Convolution

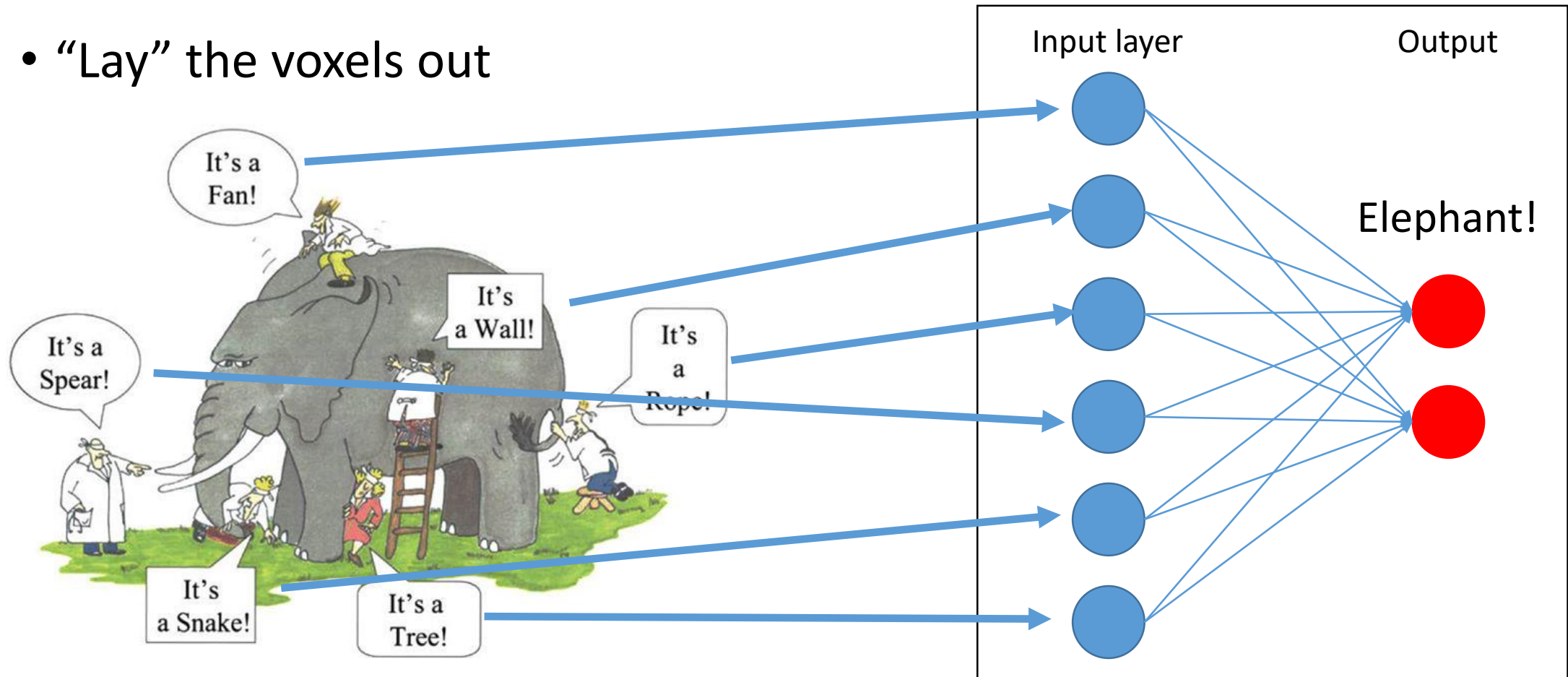
Activation

Max Pooling

Fully Connected

# Fully Connected

- “Lay” the voxels out



Convolution

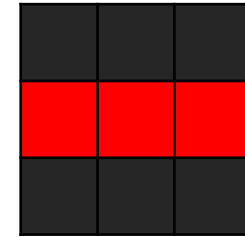
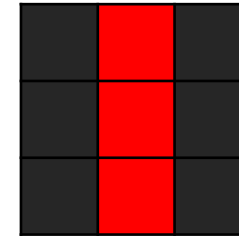
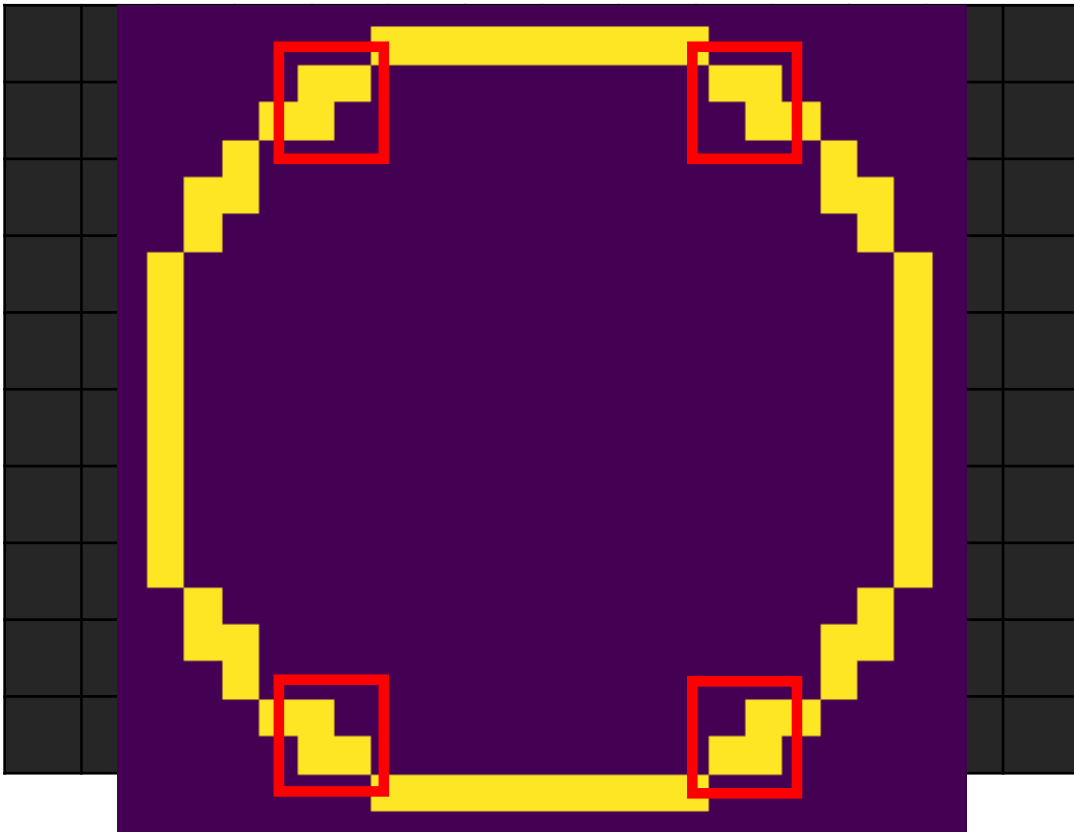
Activation

Max Pooling

Fully Connected

# Low specificity..

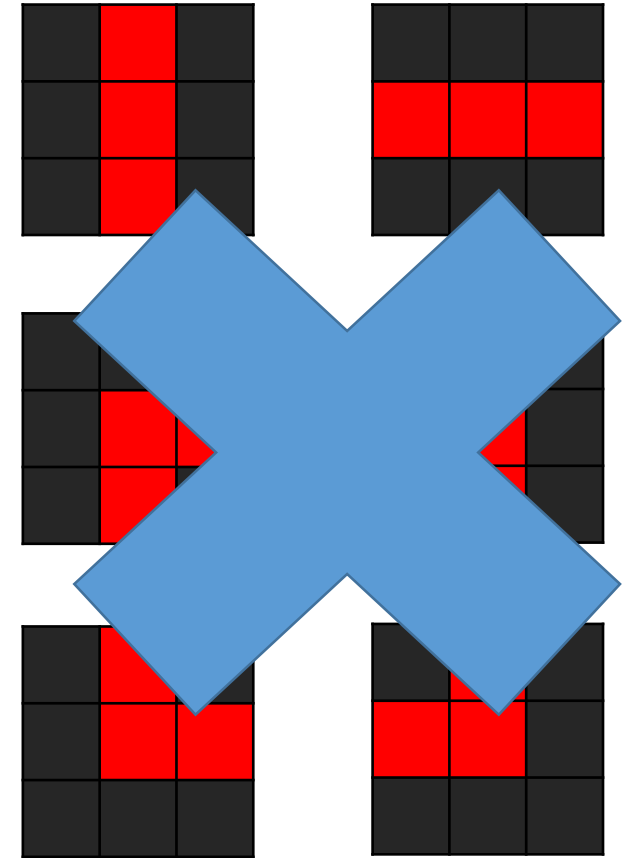
Also has two parallel lines...





# (Finally) Deep learning

- Who knows what kernels are needed!
- Why can't the stupid computer figure it out...

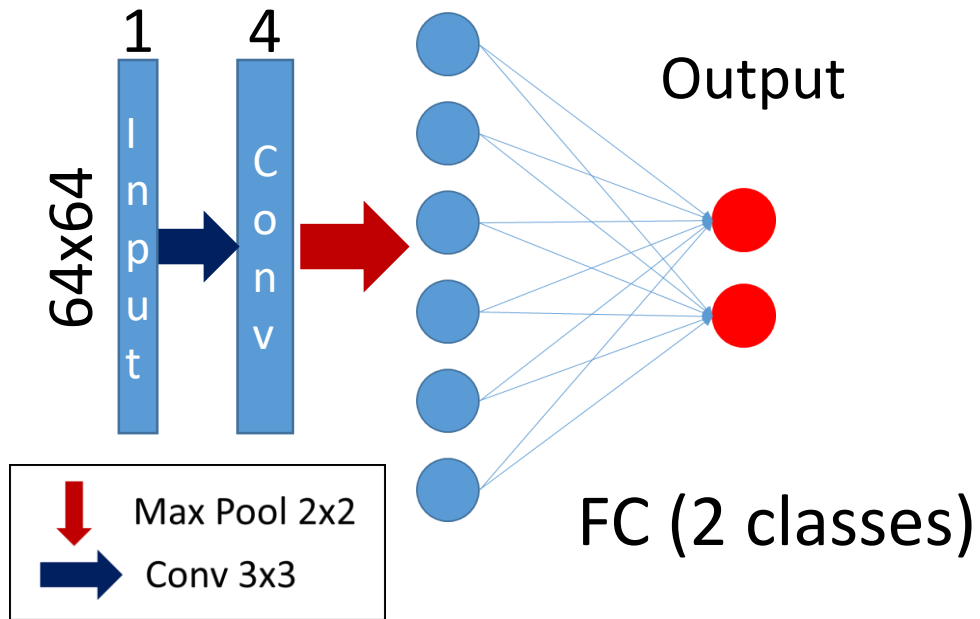


First workshop – DeepBox

# What's the difference for deep learning?

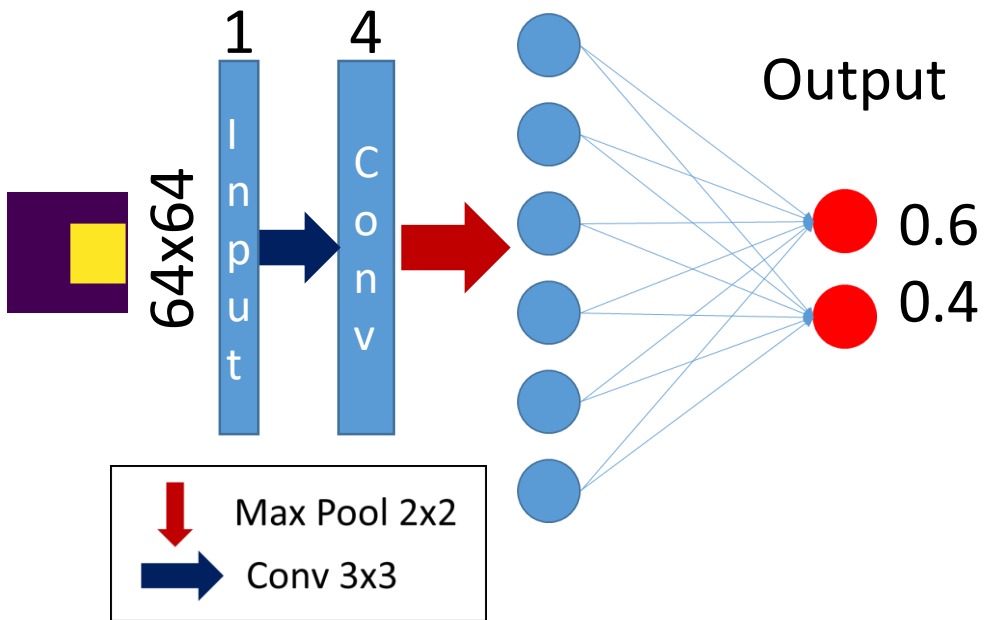
- We don't set the kernels

Random initialization



How does it learn?

# Backpropagation!

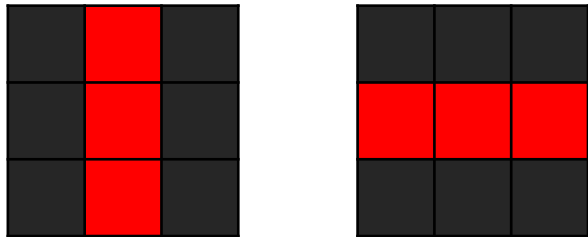


Wrong output?  
Change the decisions before it!

# Comparison

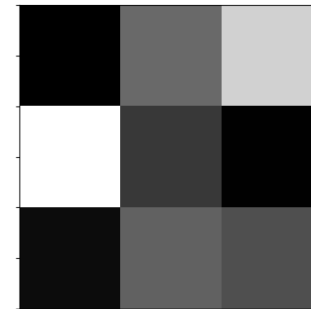
## Machine learning

- Easy to understand kernels
- Have to be clever...



## Deep learning

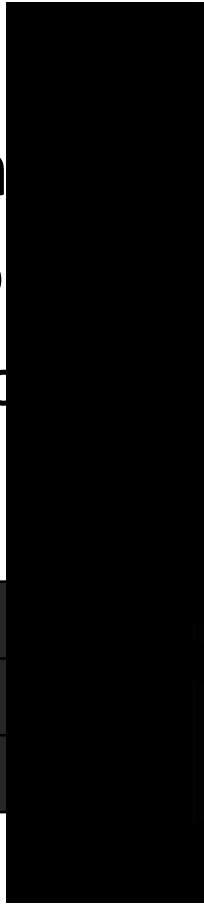
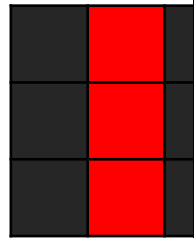
- Difficult to understand kernels
- Works great!
- Have to think about use cases
  - (Would a triangle be predicted as a rectangle or circle?)



# Comparison

## Machin

- Easy to
- Have to



When you act like you know what's you're doing 🤔



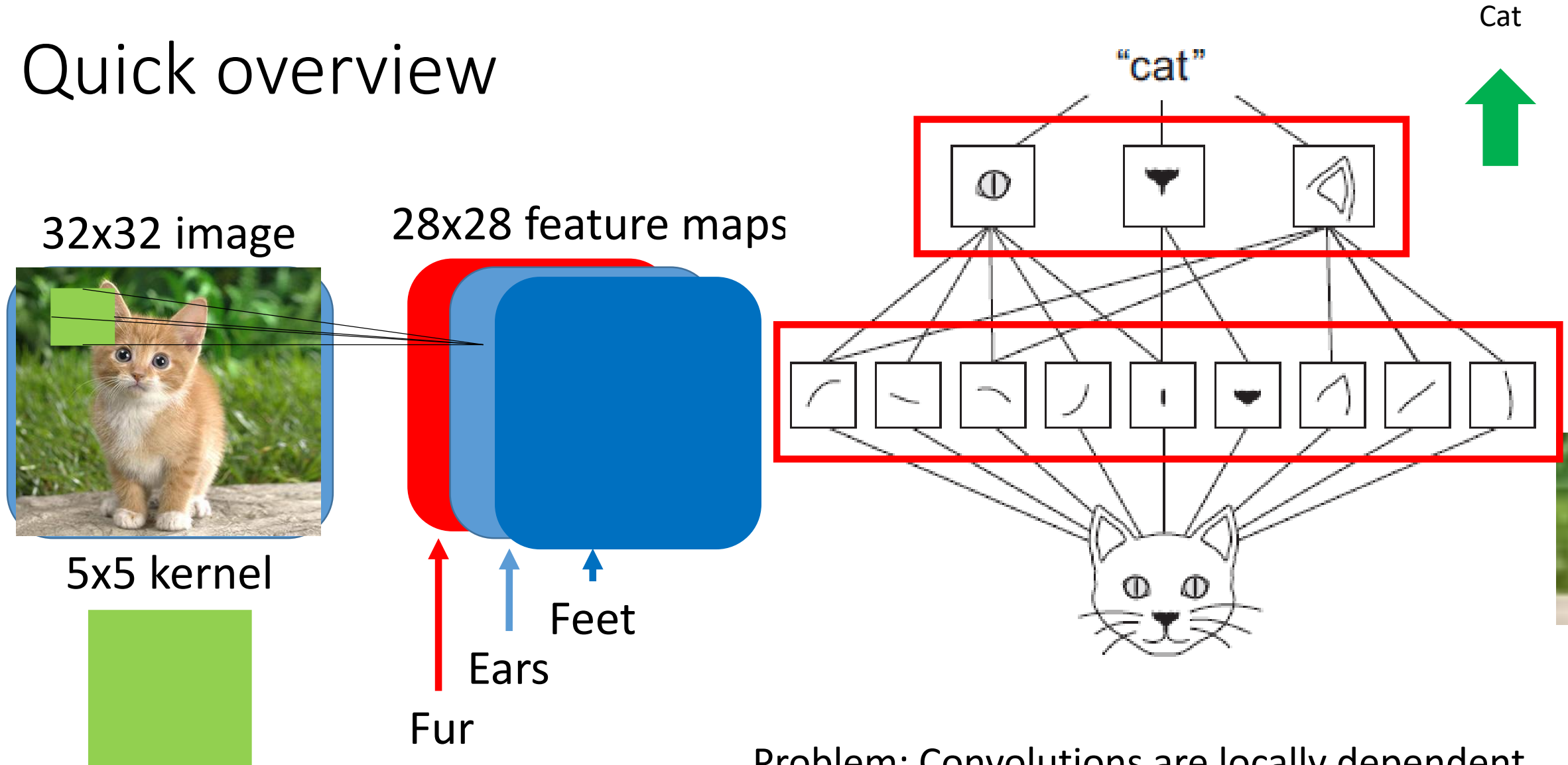
kernels

e cases

redicted as

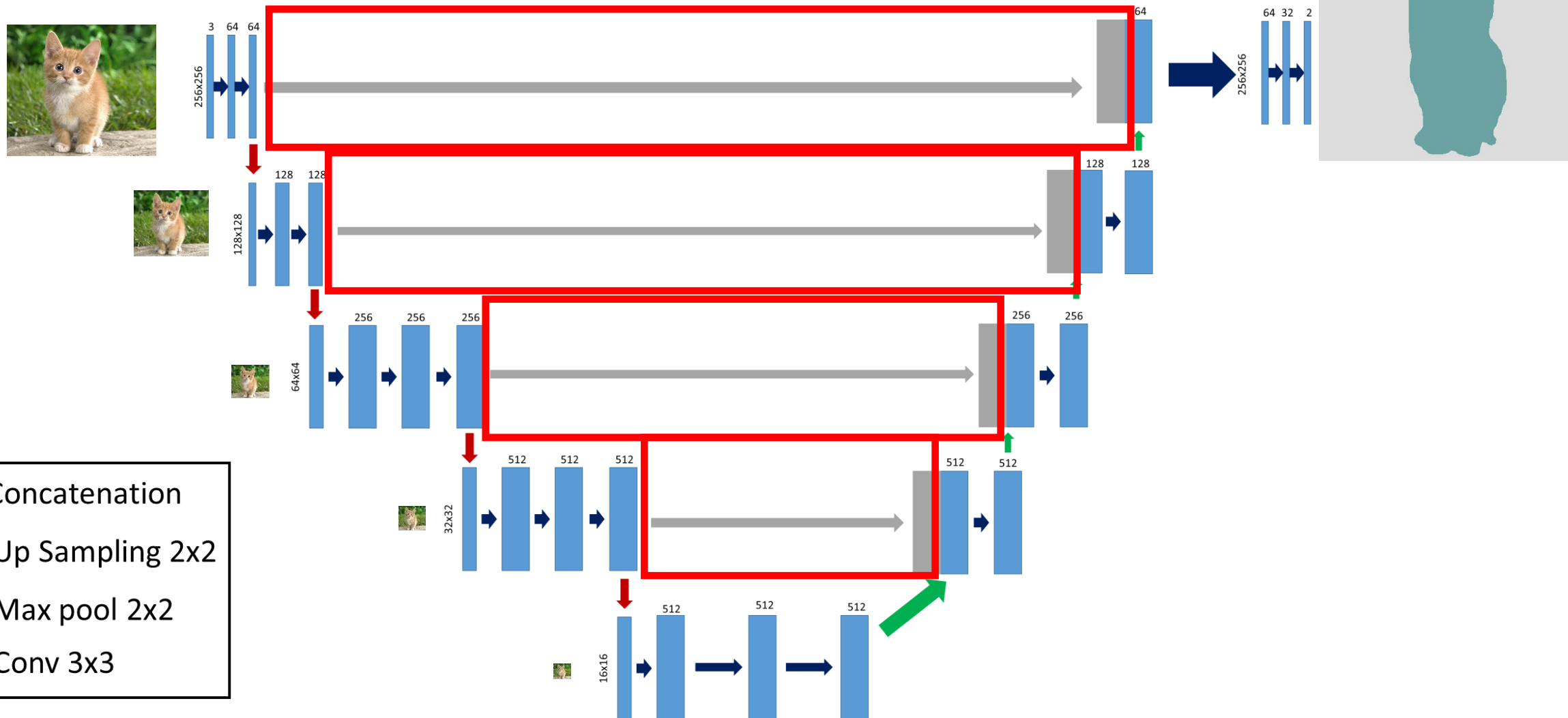
Specifically into deep learning

# Quick overview





# Auto-encoder/Decoder



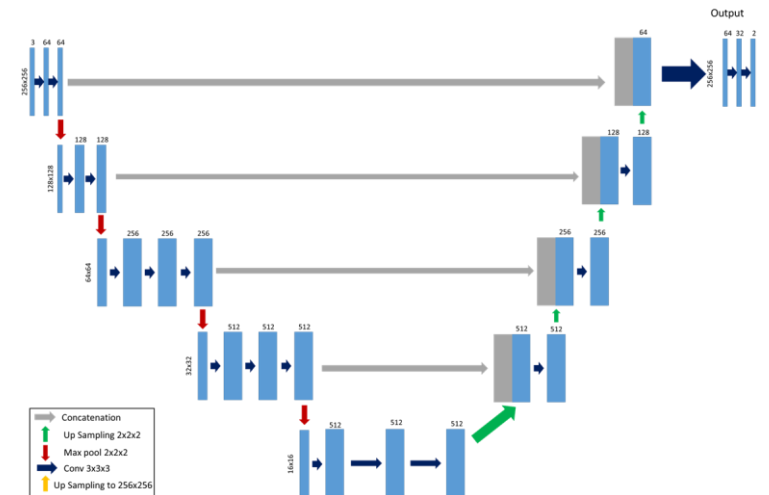
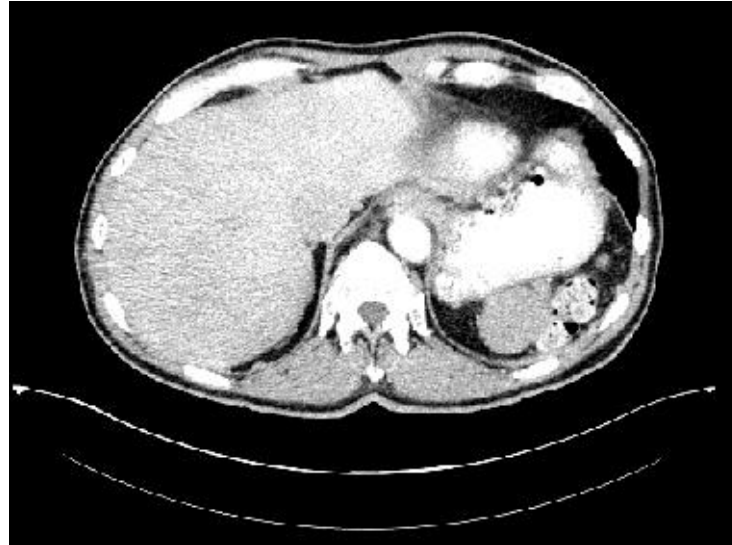
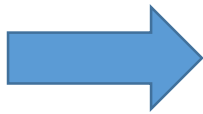
Second workshop – Liver model

# Biggest time sinks..

Roi names: Liver, liver, liver\_bma, liver\_9.15.10, etc.

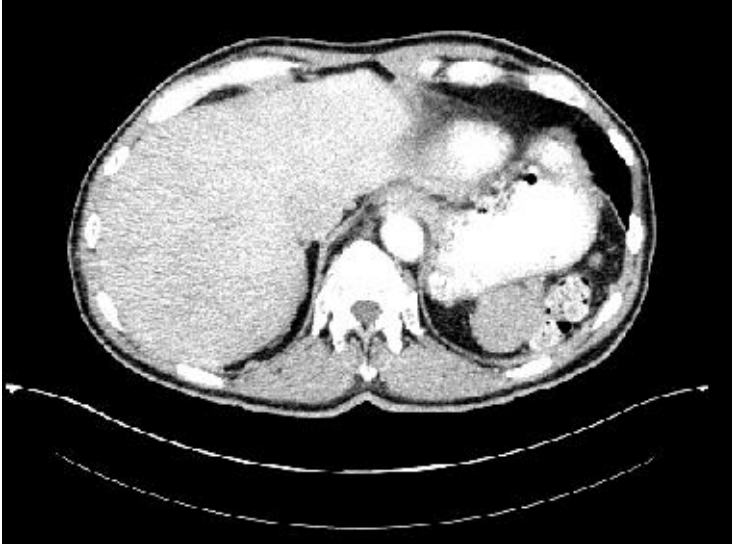
- Data curation!

- ABD\_LYMPH\_036
- ABD\_LYMPH\_069
- MED\_LYMPH\_002
- MED\_LYMPH\_017
- MED\_LYMPH\_024
- MED\_LYMPH\_029
- MED\_LYMPH\_030
- MED\_LYMPH\_044
- MED\_LYMPH\_055
- MED\_LYMPH\_063
- MED\_LYMPH\_064
- MED\_LYMPH\_067
- MED\_LYMPH\_069

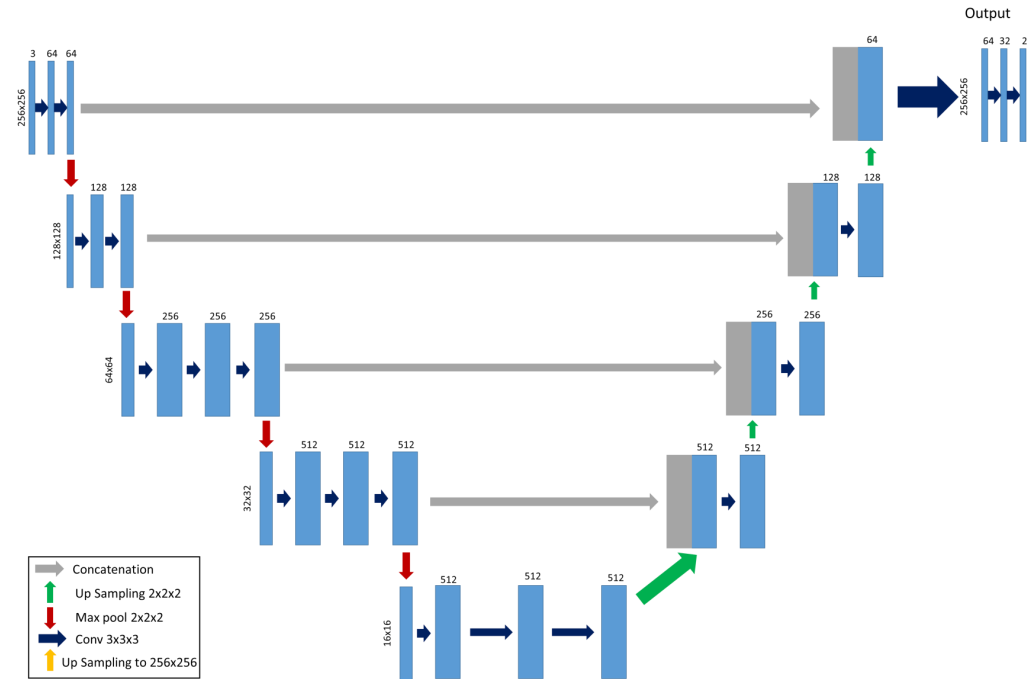


# Data Curation workbook

Now...



## Regular UNet



What about pre-trained?

# Liver Model

Workshop

# Visual Geometry Group (VGG)

# ImageNet

~15 million, 22,000 categories

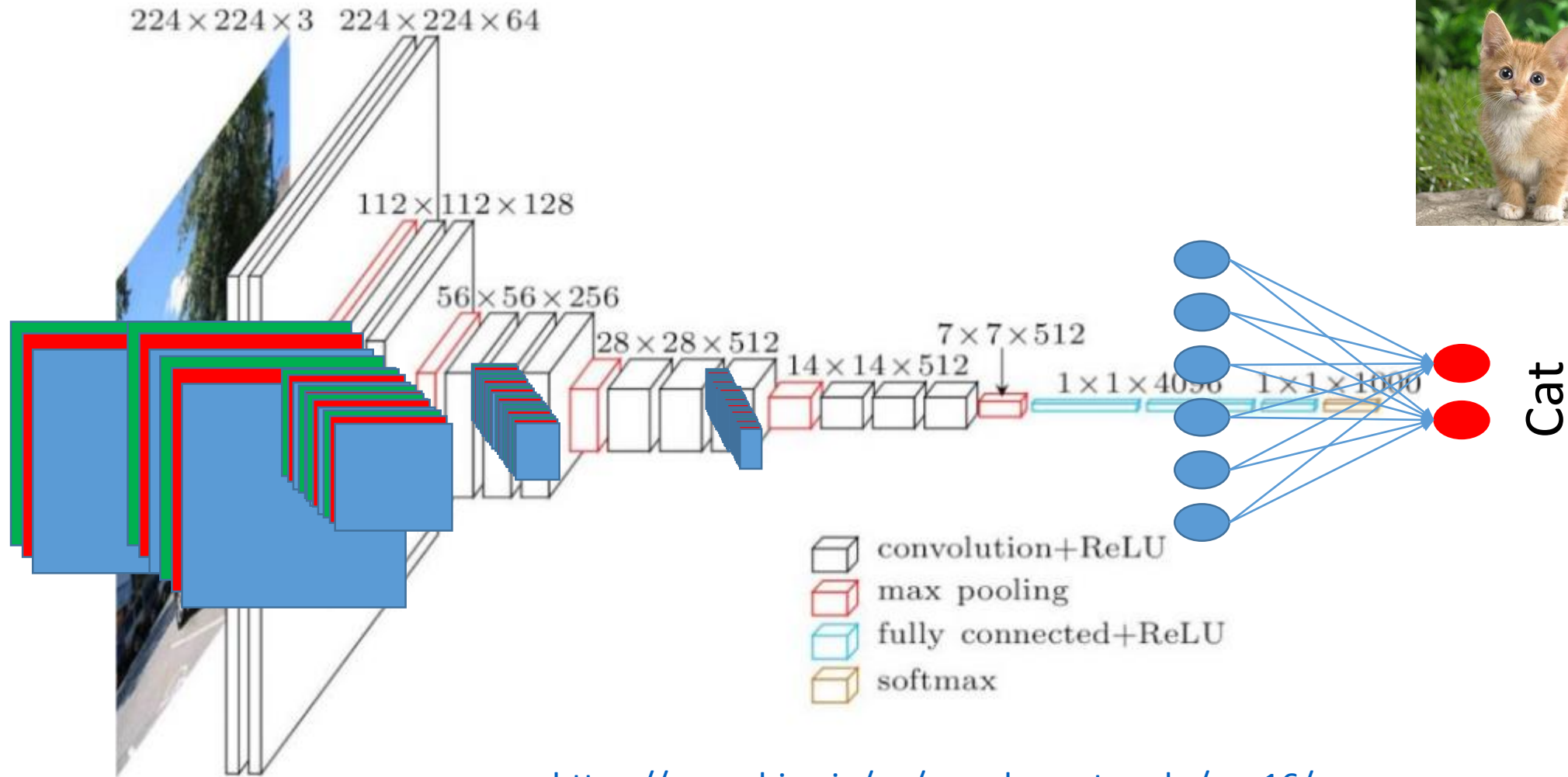
~ 1000 images in 1000 classes

Source Information	Image Distribution		
Source	Training	Validation	Test
ImageNet	1,200,000	50,000	150,000



<http://www.image-net.org/synset?wnid=n02977438>

# Visual Geometry Group (VGG)

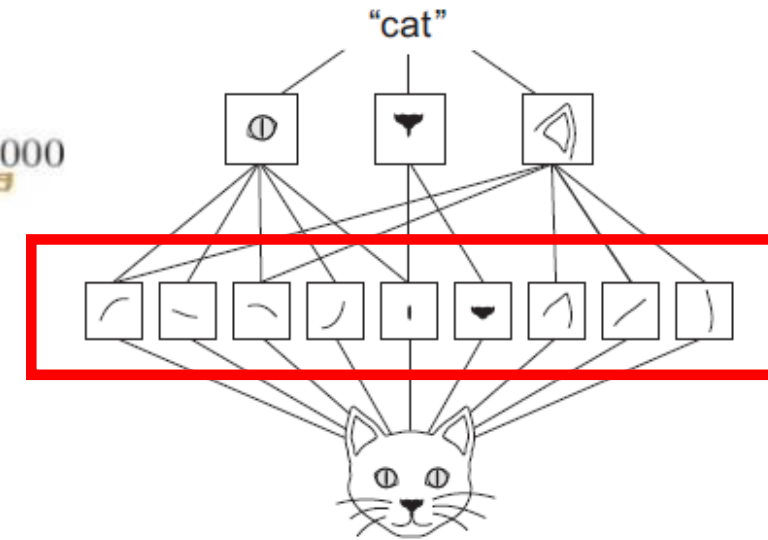
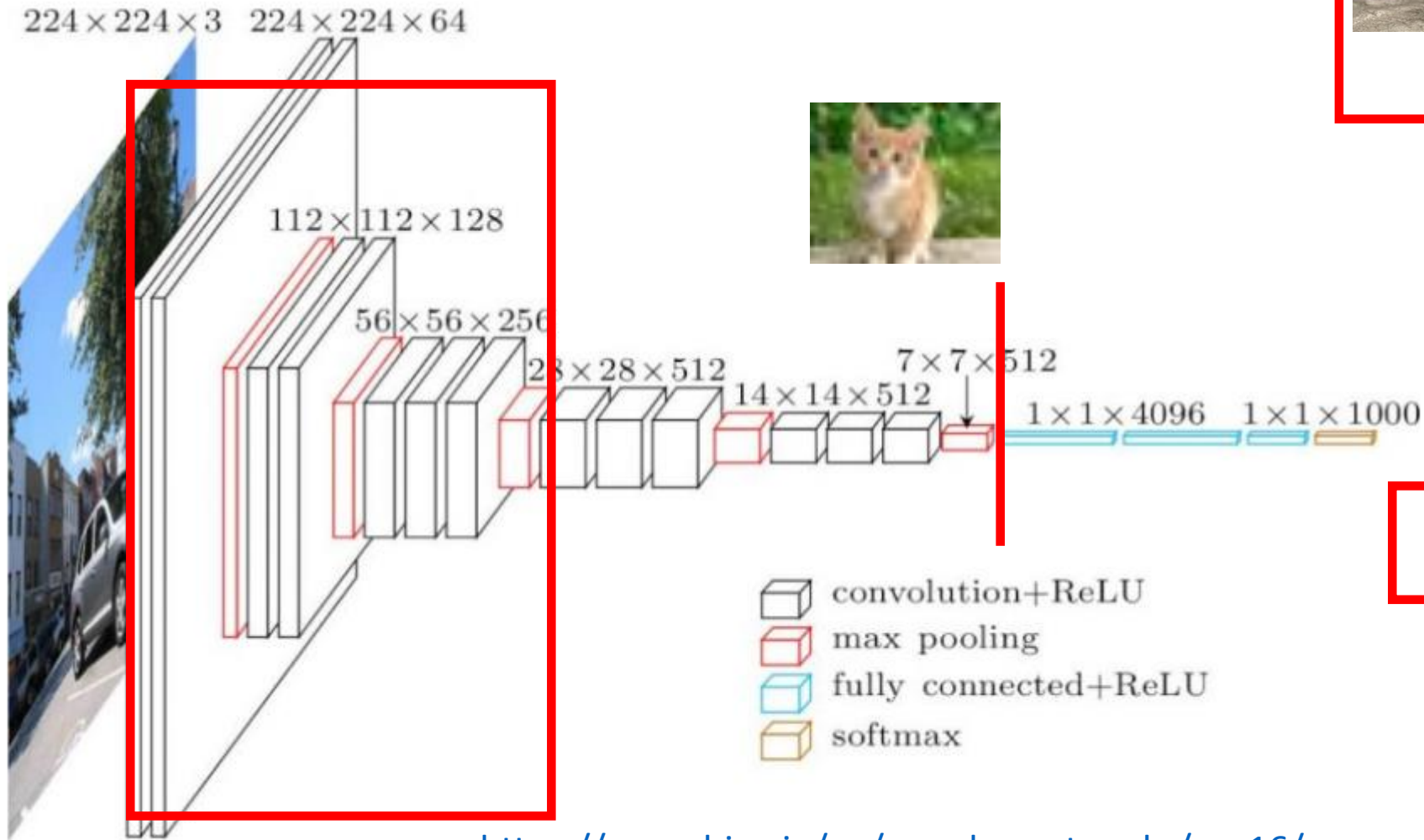


VGG16 Architecture

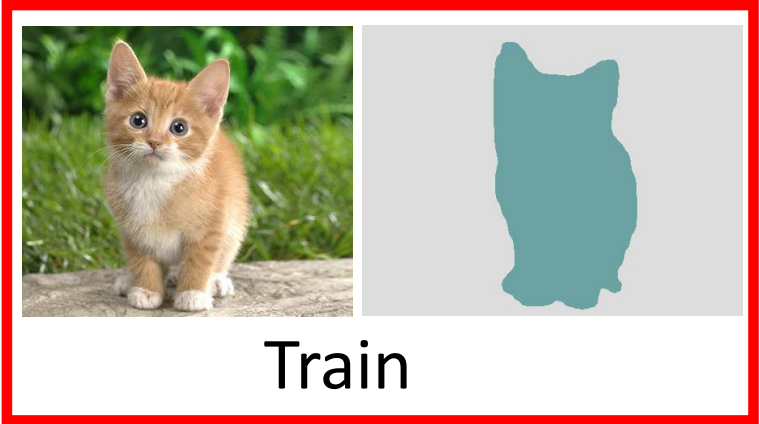
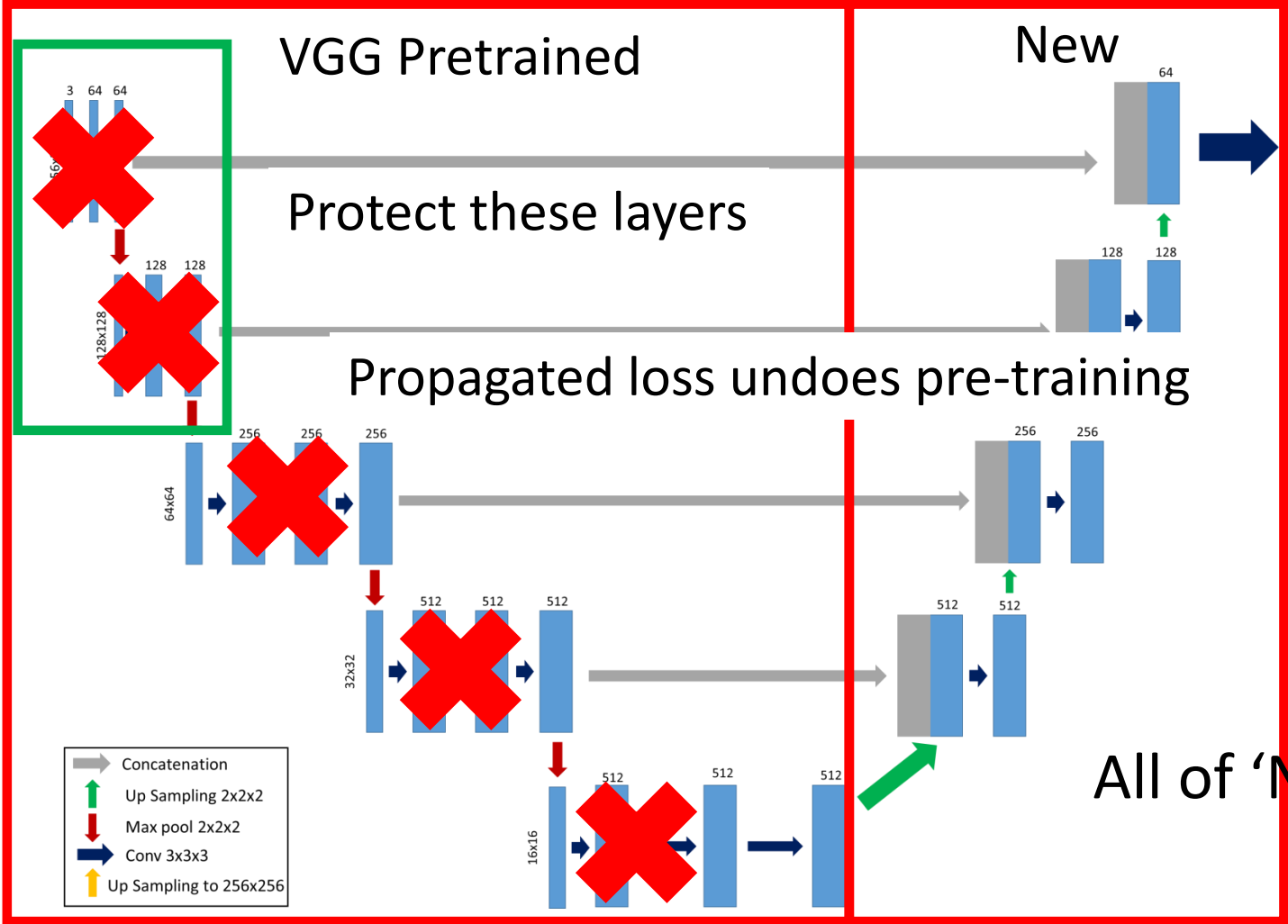
<https://neurohive.io/en/popular-networks/vgg16/>



# Visual Geometry Group (VGG)



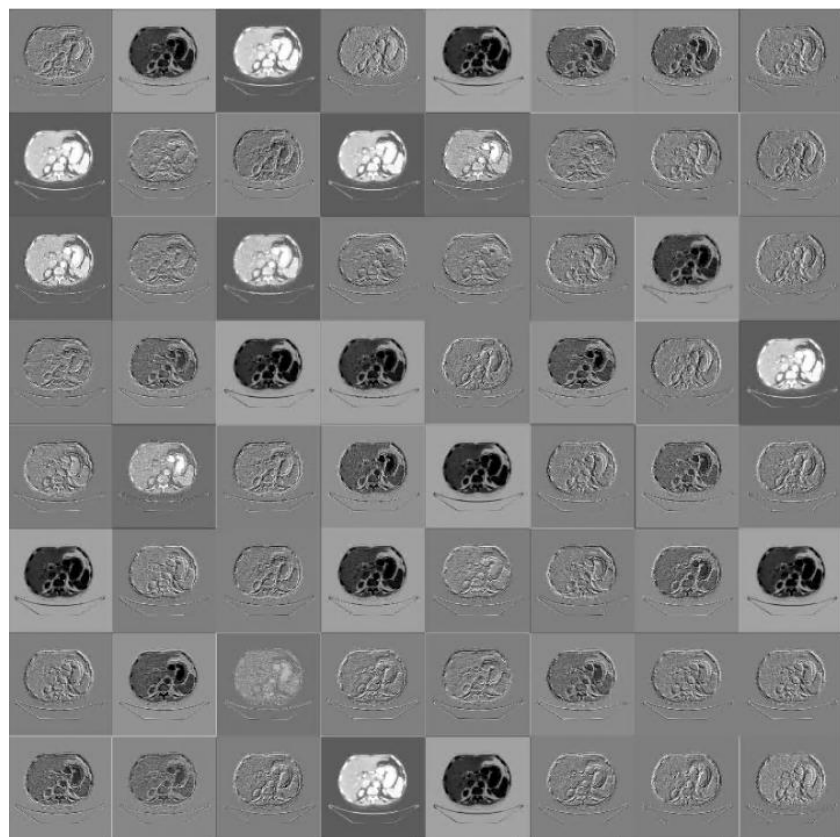
# Nuances of fine-tuning: what to re-train



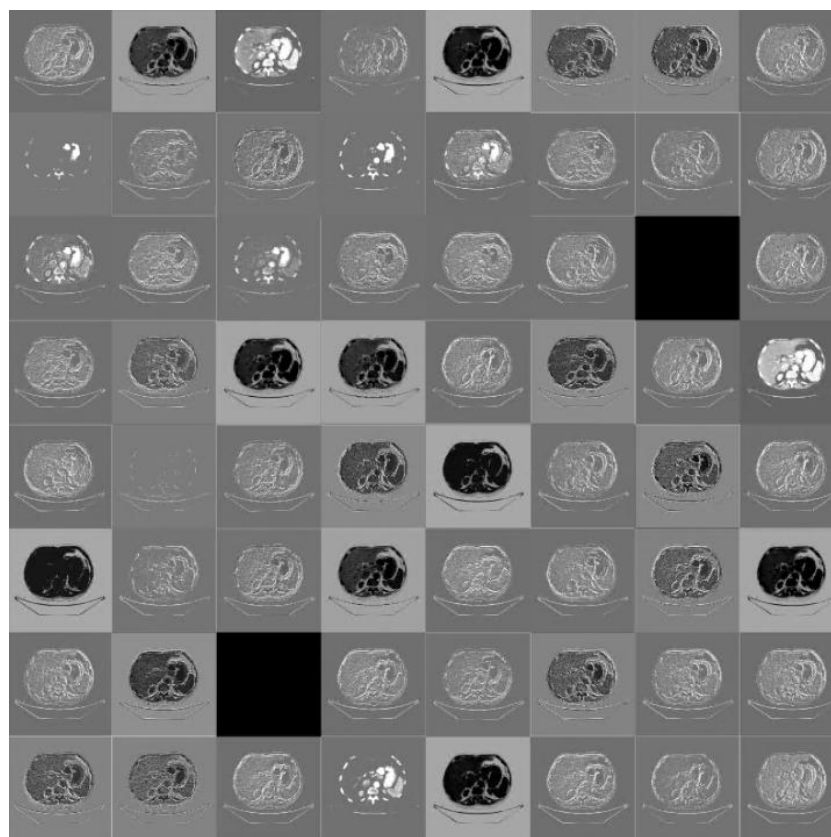
Train

# Outputs

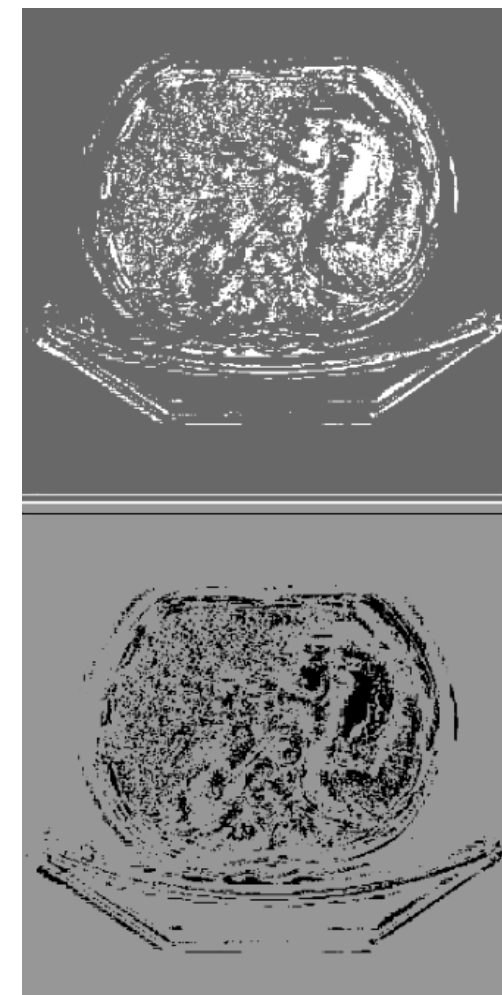
block1\_conv1



block1\_conv1\_activation



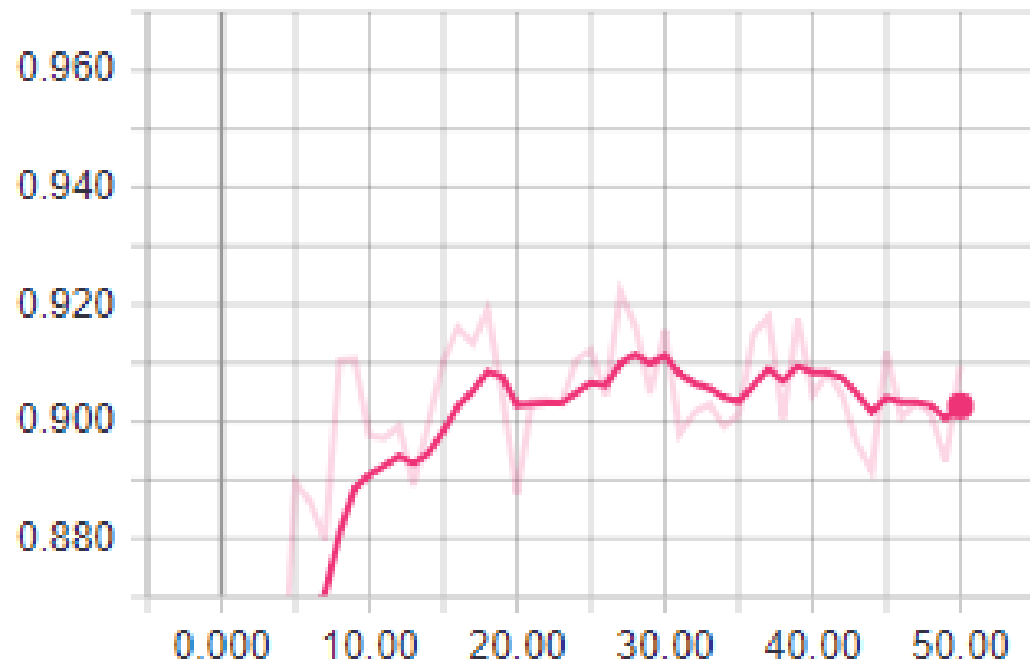
Output



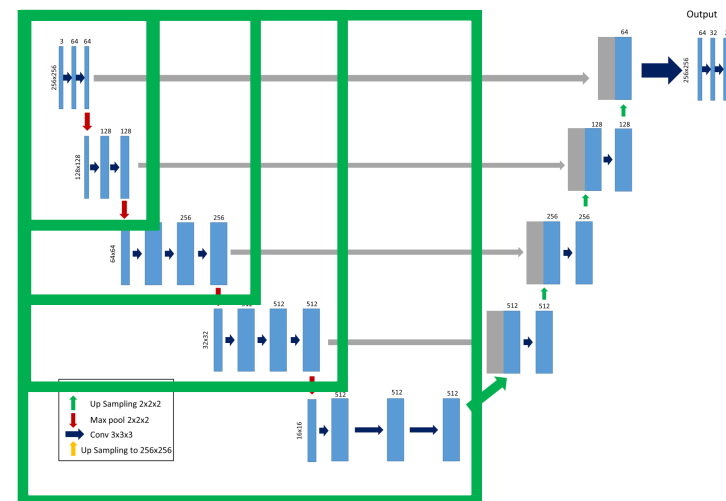
# Nuances of fine-tuning: what to re-train

What's best?

val\_dice\_coef\_3D



- VGG16 fine tune conv2 unfrozen\Tensorboard\learning rate 0.0001\run
- VGG16\_fine\_tune\_conv2\_unfrozen\_from\_start\Tensorboard\learning\_rate\_1e-05\run
- VGG16\_fine\_tune\_conv3\_unfrozen\Tensorboard\learning\_rate\_1e-05\run
- VGG16\_fine\_tune\_conv4\_unfrozen\Tensorboard\learning\_rate\_1e-05\run
- VGG16\_fine\_tune\_conv5\_unfrozen\Tensorboard\learning\_rate\_3e-05\run

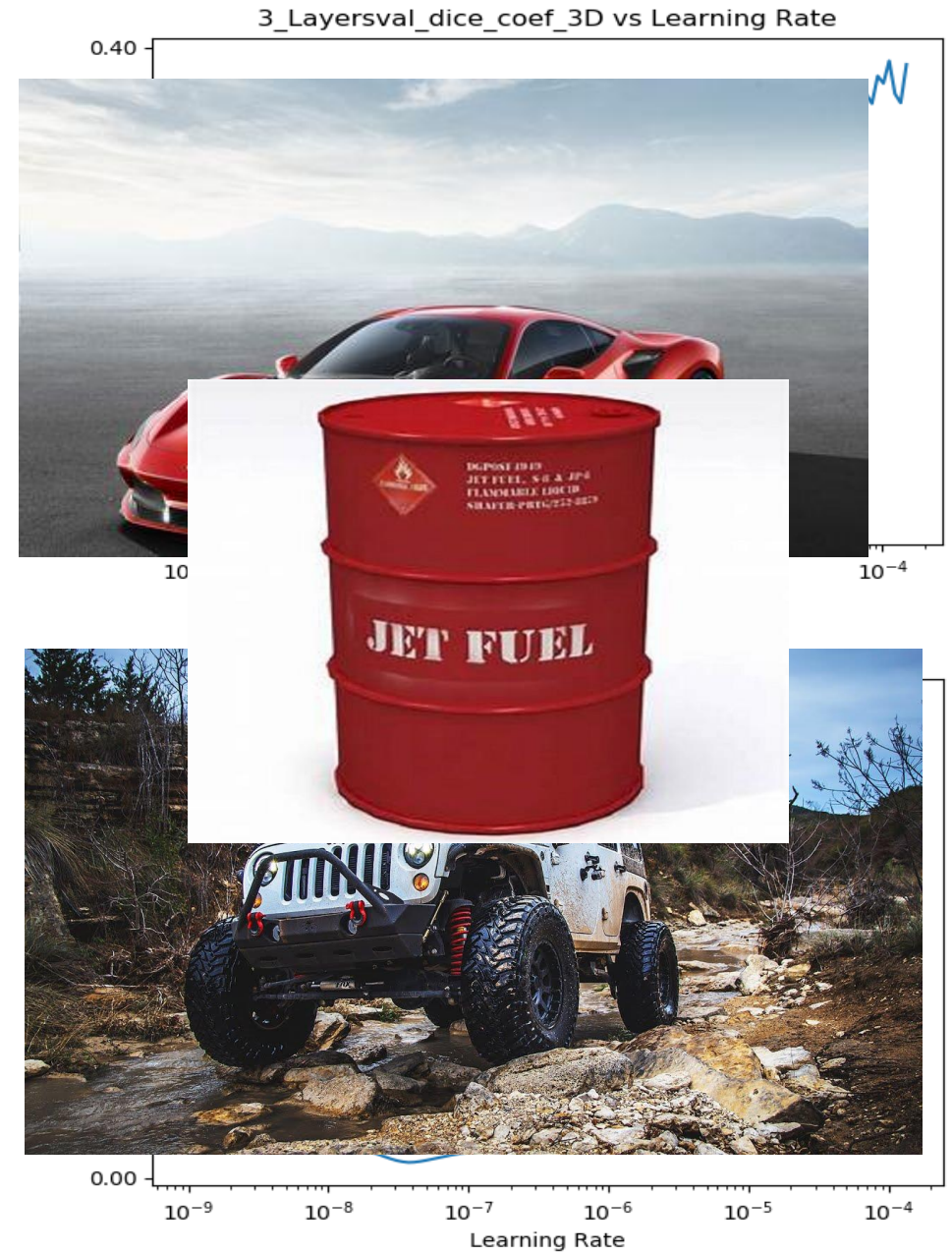


Making your own architecture

# Optimization Searching

- Things you can change...
- Architecture
  - Layers deep
  - # Convolution blocks/# Features
- Hyper-Parameters
  - Learning rate, loss, regularization

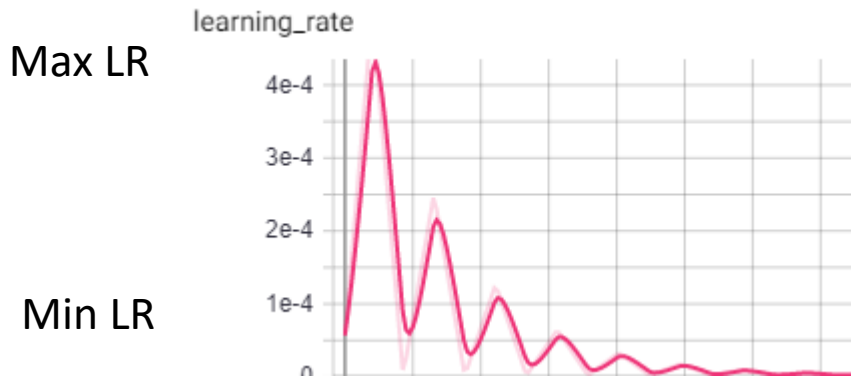
Start simple...



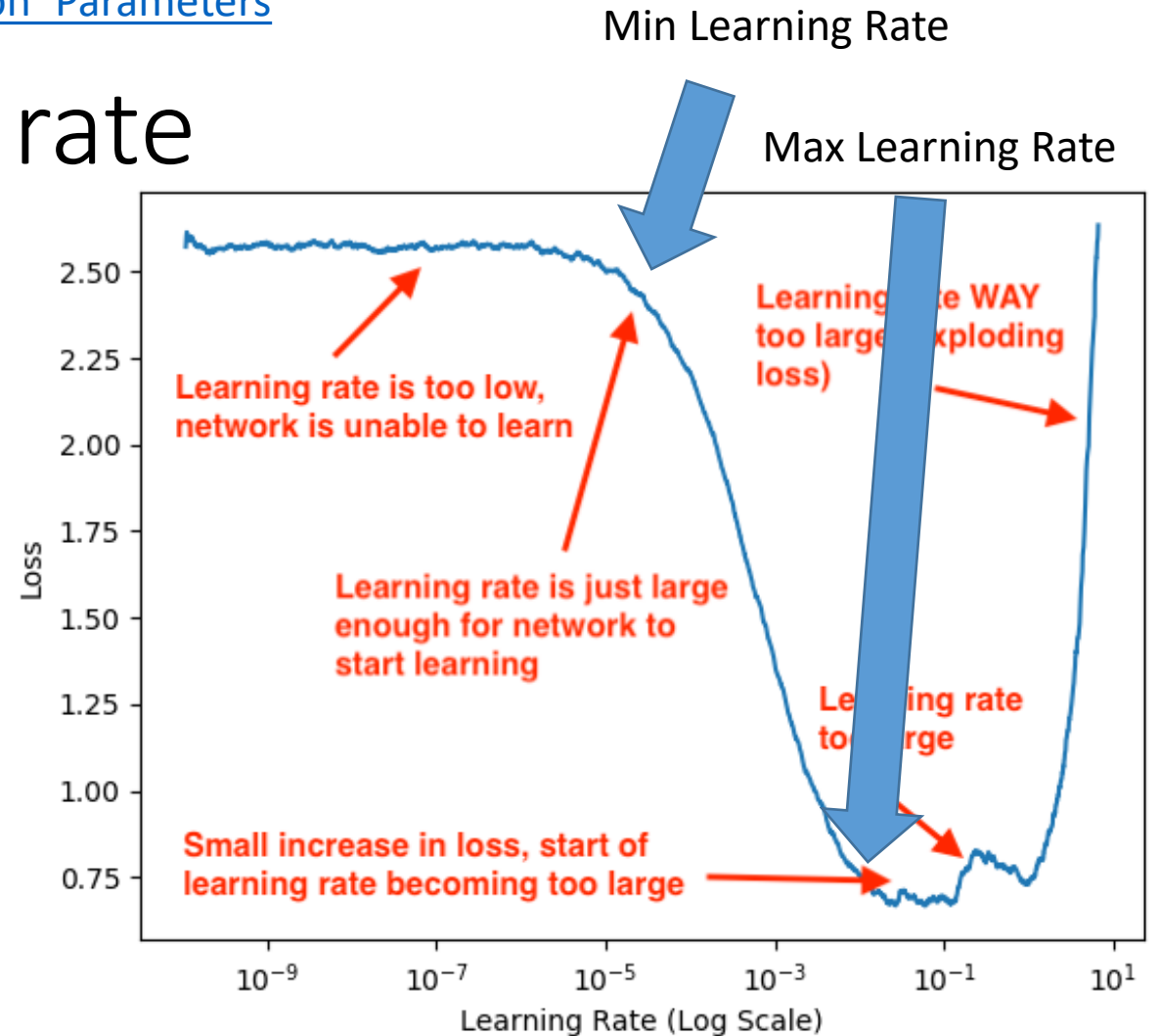
# Finding a good learning rate

- Potentially largest wastes of time...
- Gradually increase LR
  - Find min and max LR

Triangular Policy

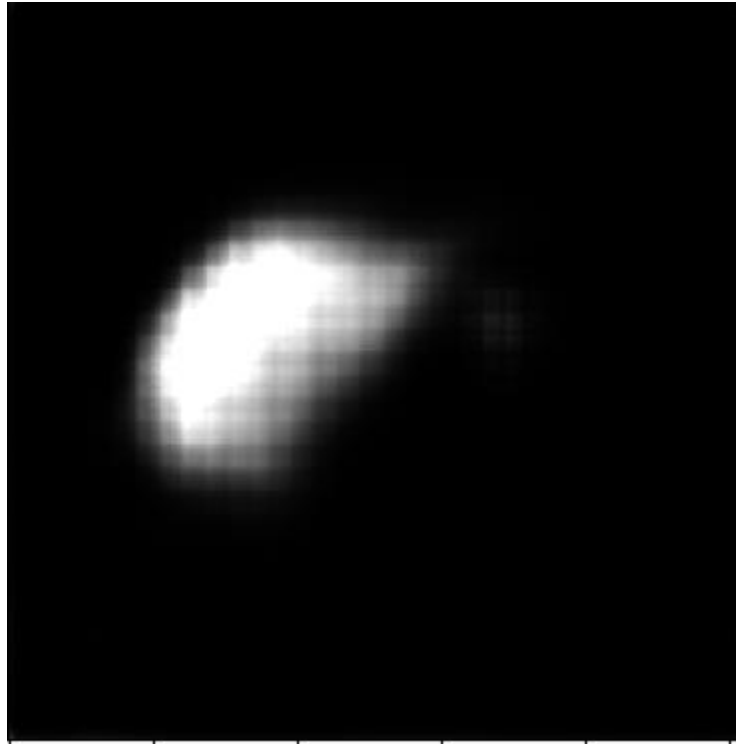


<https://arxiv.org/pdf/1506.01186.pdf>



<https://www.pyimagesearch.com/2019/08/05/keras-learning-rate-finder/>

# Inverse Transpose Artifacts

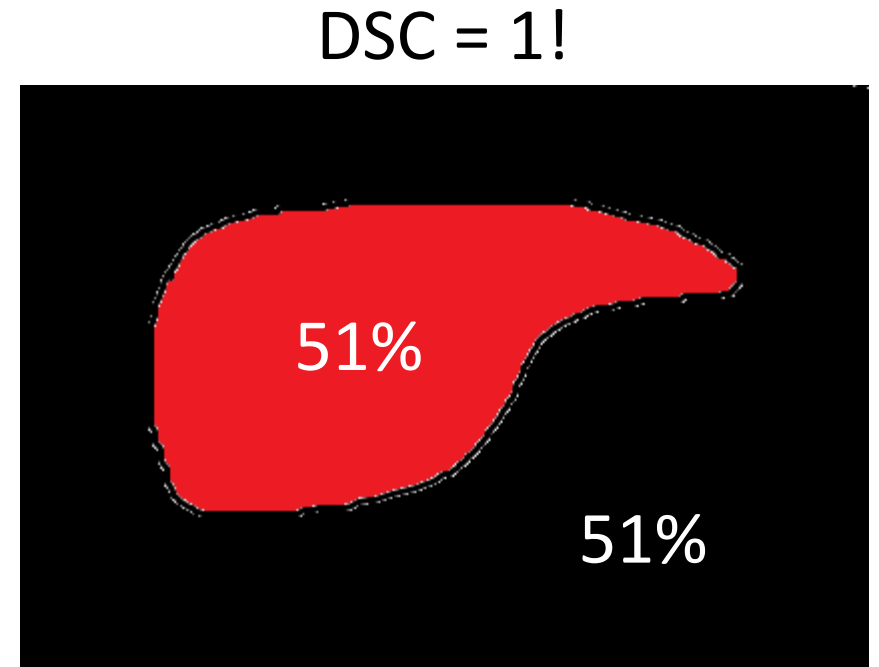




Important questions to ask

# Important questions to ask

- Evaluating final algorithm
  - Using same loss as before? Dice?
- What is your loss metric?
  - Dice, Categorical cross entropy



Doesn't mean Dice is bad!  
Need other metrics with it

# Important questions to ask



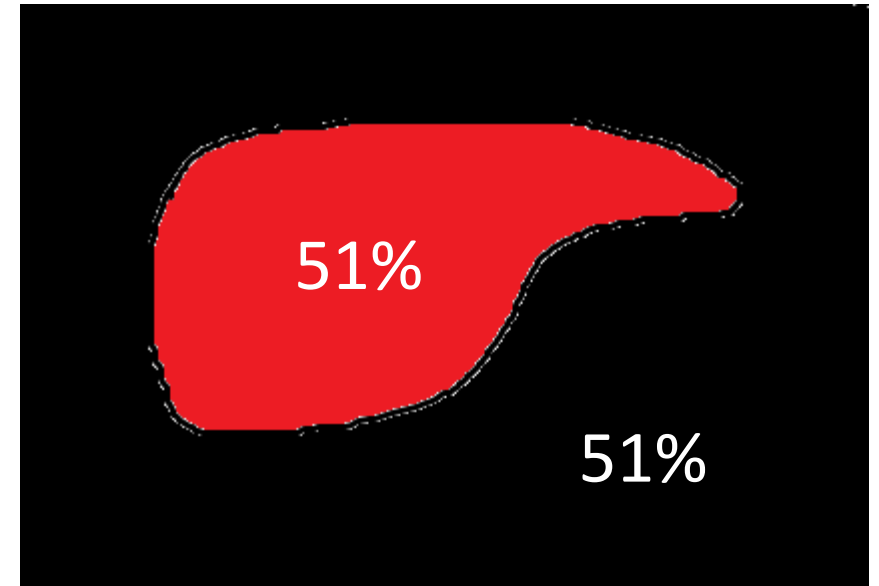
2 Classes  
Boat  
Not-Boat



# Important questions to ask

- Evaluating final algorithm
  - Using same loss as before? Dice?
- What is your loss metric?
  - Dice, Categorical cross entropy
- WITHHELD TEST SET?!
- Qualitative assessment?

DSC = 1!

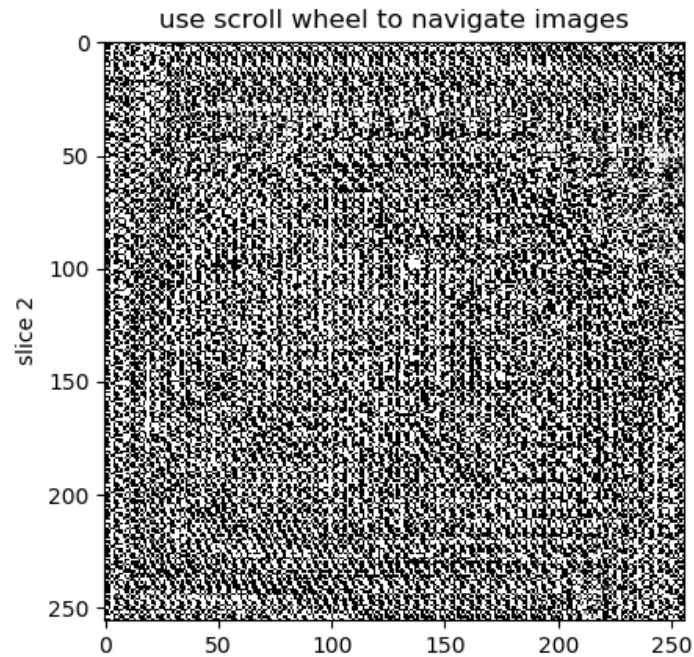


Doesn't mean Dice is bad!  
Need other metrics with it



# Oops

- Accuracy: 0.92
- Prediction



```
drop6 = Dropout(0.6)(conv6)
up7 = Conv3D(256, filters, activation='relu', padding='same', kernel_initializer='he_normal')(
    |   UpSampling3D(size=(2, 2, 2))(drop6)
# merge7 = merge([crop3, up7], mode='concat', concat_axis=3)
merge7 = concatenate([conv3, up7], axis=3)
conv7 = Conv3D(256, filters, activation='relu', padding='same', kernel_initializer='he_normal')(merge7)
conv7 = Conv3D(256, filters, activation='relu', padding='same', kernel_initializer='he_normal')(conv7)

drop7 = Dropout(0.8)(conv7)
up8 = Conv3D(128, filters, activation='relu', padding='same', kernel_initializer='he_normal')(
    |   UpSampling3D(size=(2, 2, 2))(drop7)
# merge8 = merge([crop2, up8], mode='concat', concat_axis=3)
merge8 = concatenate([conv2, up8], axis=3)

conv8 = Conv3D(128, filters, activation='relu', padding='same', kernel_initializer='he_normal')(merge8)
conv8 = Conv3D(128, filters, activation='relu', padding='same', kernel_initializer='he_normal')(conv8)

drop8 = Dropout(0.9)(conv8)

up9 = Conv3D(64, filters, activation='relu', padding='same', kernel_initializer='he_normal')(
    |   UpSampling3D(size=(2, 2, 2))(drop8)
# merge9 = merge([crop1, up9], mode='concat', concat_axis=3)
merge9 = concatenate([conv1, up9], axis=
conv9 = Conv3D(64, filters, activation='
', padding='same', kernel_initializer='he_normal')(merge9)
conv9 = Conv3D(64, filters, activation=
', padding='same', kernel_initializer='he_normal')(conv9)

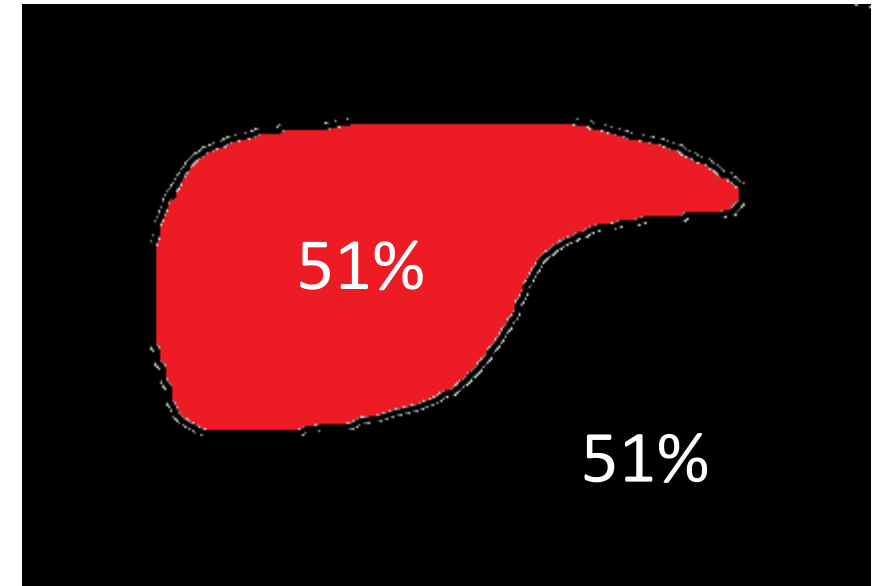
output = Conv3D(2, filters, activation='relu', padding='same', kernel_initializer='he_normal')(conv9)

output = Activation('softmax')(output)
# model = Model(inputs=inputs, outputs=output)
model = Model(inputs=[inputs], outputs=[output])
```

# Important questions to ask

- Evaluating final algorithm
  - Using same loss as before? Dice?
- What is your loss metric?
  - Dice, Categorical cross entropy
- WITHHELD TEST SET?!
- Qualitative assessment?
- What optimizer are you using?
  - Adam, SGD, (maybe RAdam)

DSC = 1!



Doesn't mean Dice is bad!  
Need other metrics with it



# Acknowledgements and Thanks

## Brock Lab Group

Kristy Brock, PhD  
Molly McCulloch  
Guillaume Cazoulat, PhD  
Bastien Rigaud  
Yulun He  
Andrea Ohrt, MBA  
Alexandra Ford  
Jason King  
Anando Sen, PhD  
Anne-Cecile Lesage, PhD

Eugene Koay, MD

## Odisio Lab Group

Bruno Odisio, MD  
Ethan Lin, MD

## Committee

Kristy Brock, PhD  
Laurence E Court, PhD  
Erik N.K. Cressman, MD, PhD  
Ankit B Patel, PhD  
Carlos C Cardenas, PhD

## Court Yard Group

Laurence Court, PhD  
Kelly Kisling, MS  
Carlos Cardenas, PhD  
Tucker Netherton, DMP  
Rachel McCarroll, PhD  
Jinzhong Yang, PhD  
Luciano Branco  
Rachel Ger  
Skylar Gay  
Casey Gay  
Joy Zhang, PhD  
Callistus Nguyen, PhD  
Constance Owens  
Dong Joo Rhee, MS  
Yvonne Roed, PhD

**Thank you!**

[www.Github.com/BrianMAnderson](http://www.Github.com/BrianMAnderson)

[bmanderson@mdanderson.org](mailto:bmanderson@mdanderson.org)

- <https://media.giphy.com/media/st83jeYy9L6Bq/giphy.gif> (peter throwing blinds 'tweaking neural network')
- <https://www.youtube.com/watch?v=bnJ8UpvdTQY> (kid can't say animal names)