Convolutional Neural Networks

Executive-ML 2017/09/21

Bubacarr Bah
AIMS South Africa, & Stellenbosch University

Adapted from Alex Conway

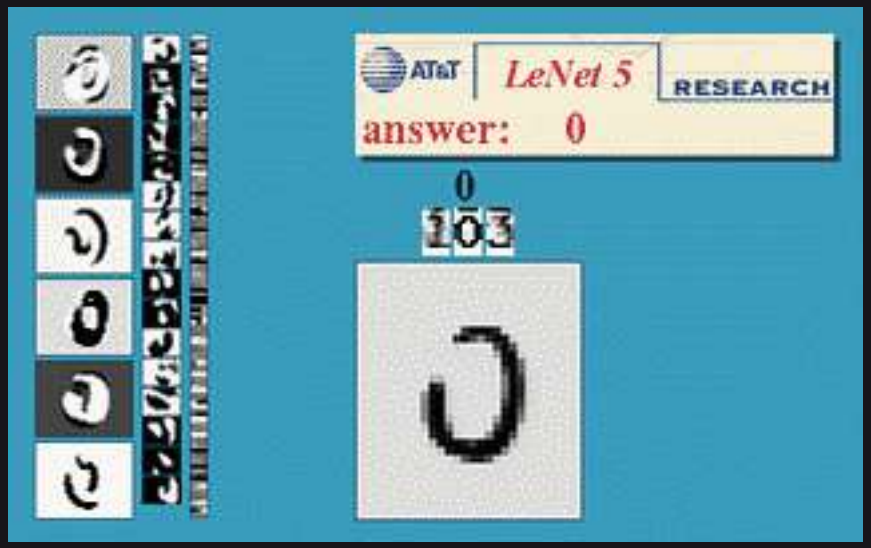
Check out the
Deep Learning Indaba
videos & practicals!

http://www.deeplearningindaba.com/videos.html
http://www.deeplearningindaba.com/practicals.html

CNNs a very powerful tool for Computer Vision

- Image classification
- Object detection
- Image captioning & visual attention
- Image Q&A, and video Q&A
- Pix2Pix
- Style transfer
- …

Image Classification



http://yann.lecun.com/exdb/mnist/

https://github.com/fchollet/keras/blob/master/examples/mnist_cnn.py
(99.25% test accuracy in 192 seconds and 70 lines of code)

Image Classification

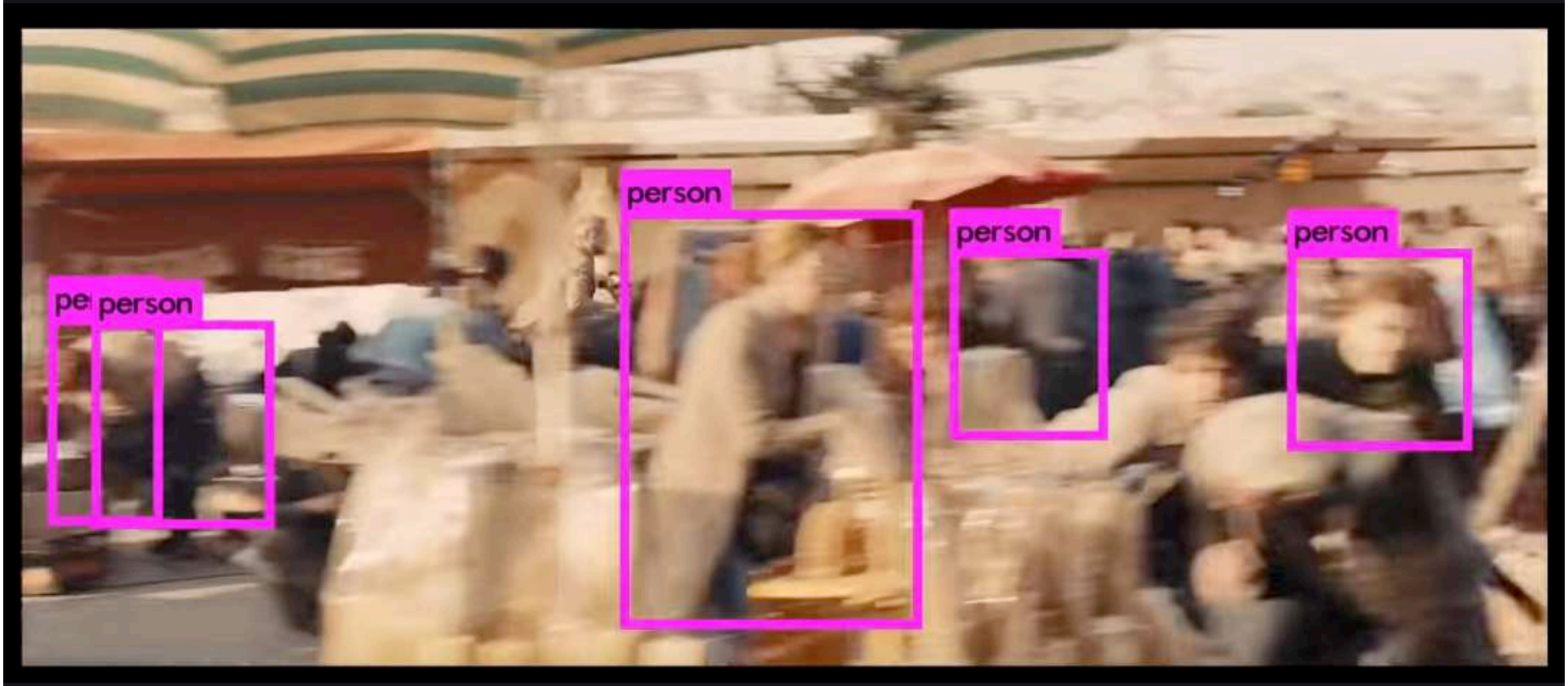
Image Classification

ImageNet Classification with Deep Convolutional Neural Networks, Krizhevsky et. Al. Advances in Neural Information Processing Systems 25 (NIPS2012)

Object detection

https://research.googleblog.com/2017/06/supercharge-your-computer-vision-models.html

Object detection



https://www.youtube.com/watch?v=VOC3huqHrss

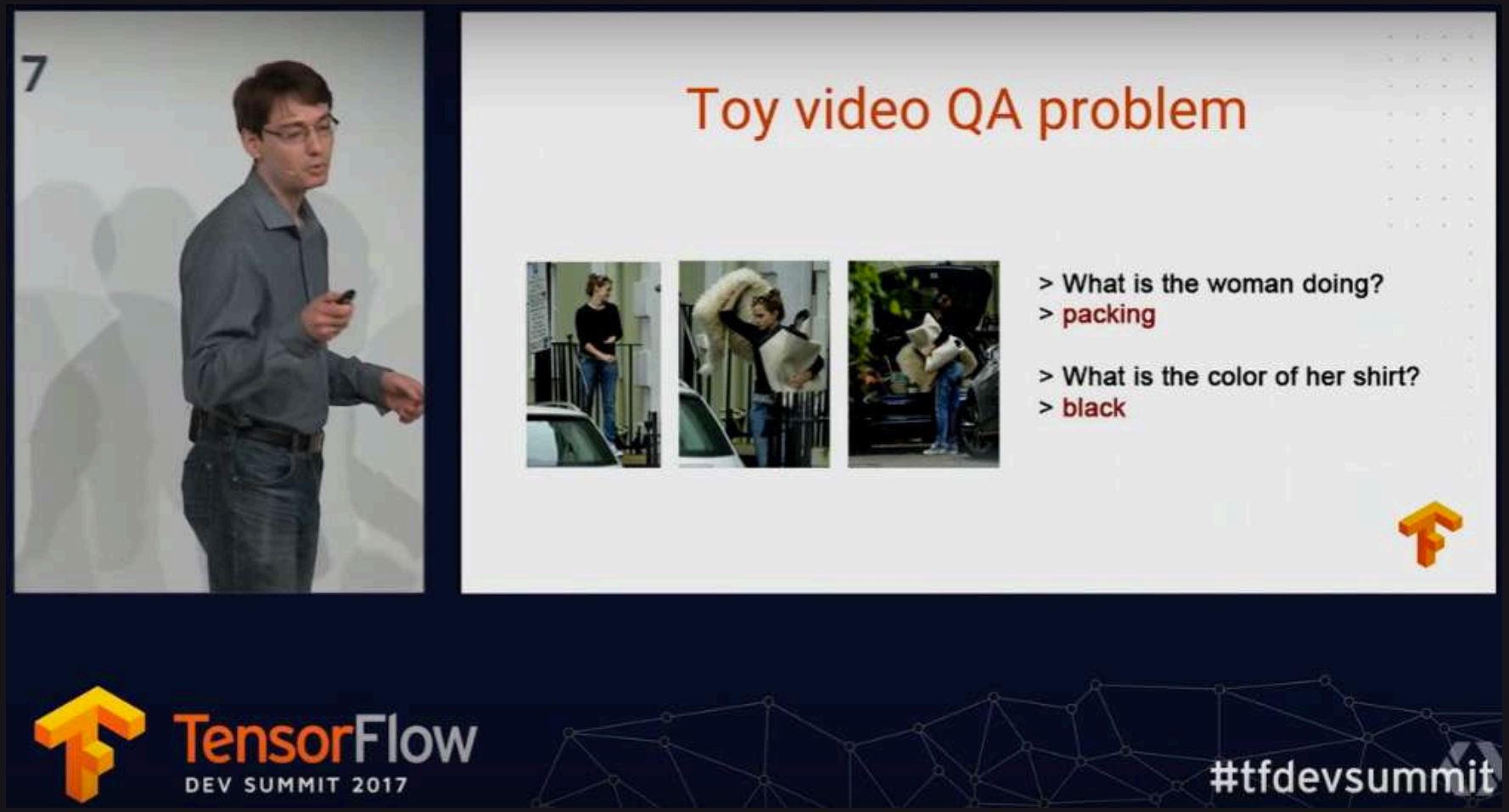
Image Captioning & Visual Attention

https://einstein.ai/research/knowing-when-to-look-adaptive-attention-via-a-visual-sentinel-for-image-captioning

Image Q&A

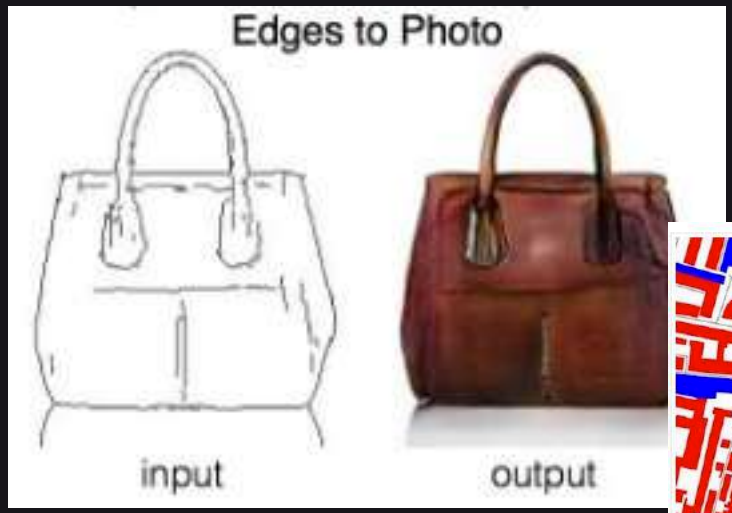
https://arxiv.org/pdf/1612.00837.pdf

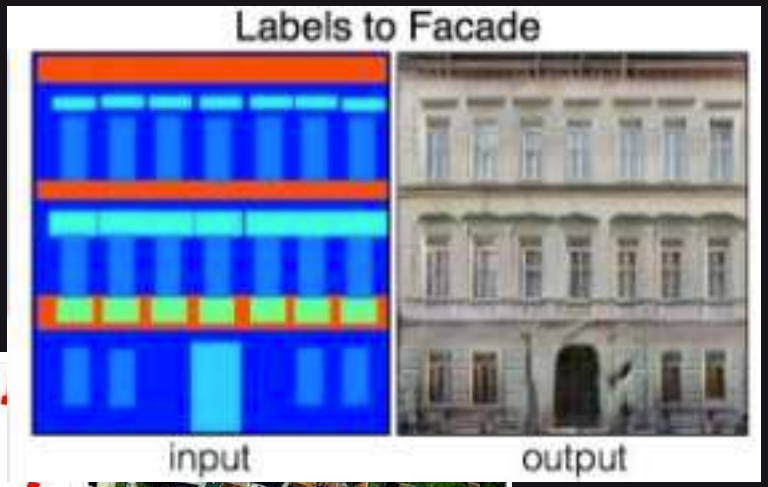
Video Q&A



https://www.youtube.com/watch?v=UeheTiBJ0Io

Pix2Pix





https://affinelayer.com/pix2pix/

https://github.com/affinelayer/pix2pix-tensorflow

Pix2Pix

https://medium.com/towards-data-science/face2face-a-pix2pix-demo-that-mimics-the-facial-expression-of-the-german-chancellor-b6771d65bf66

Original input

Rear Window (1954)

Pix2pix output

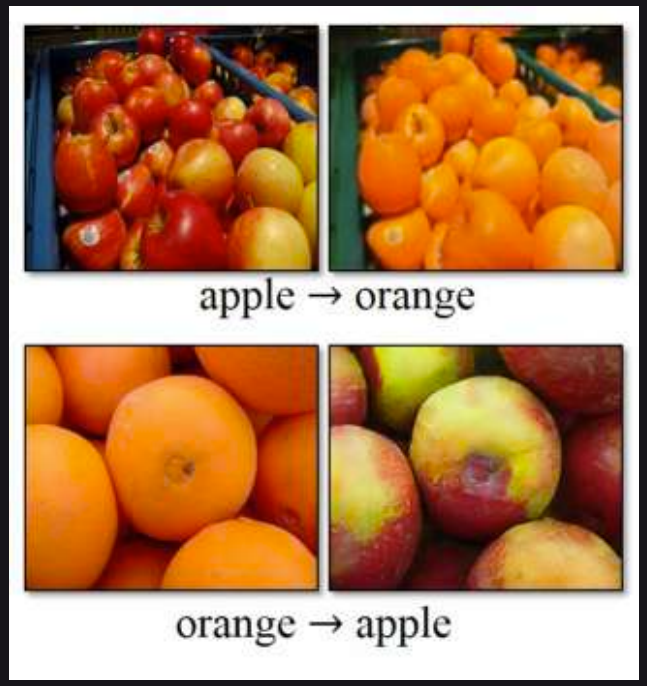
Fully Automated

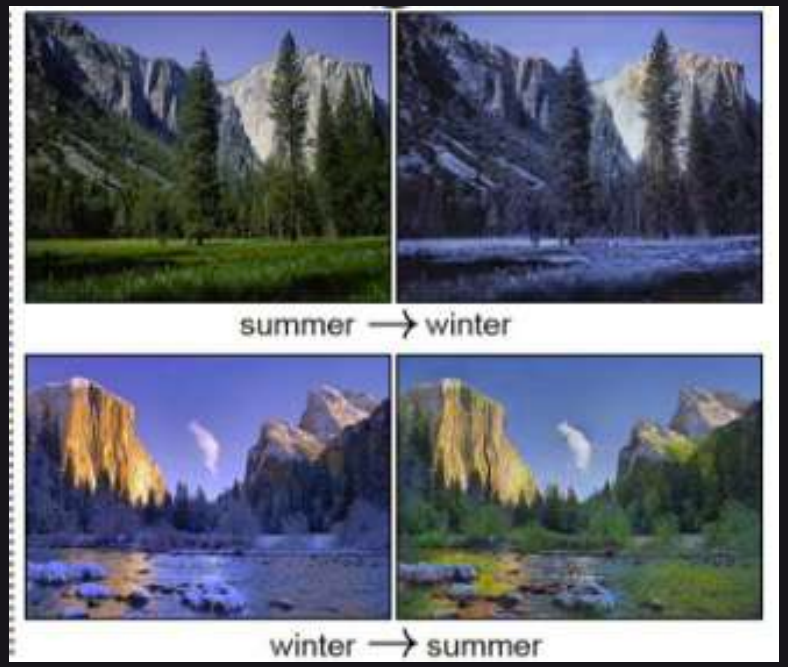
Remastered

Painstakingly by Hand

https://hackernoon.com/remastering-classic-films-in-tensorflow-with-pix2pix-f4d551fa0503

Style Transfer





https://github.com/junyanz/CycleGAN

~~Style~~ ~~Transfer~~ MAGIC

https://github.com/junyanz/CycleGAN

Real Fake News

https://www.youtube.com/watch?v=MVBe6_o4cMI

1. What is a neural network?
2. What is a convolutional neural network?
3. How to use a convolutional neural network
4. More advanced Methods
5. Case studies & applications

Big Shout Outs

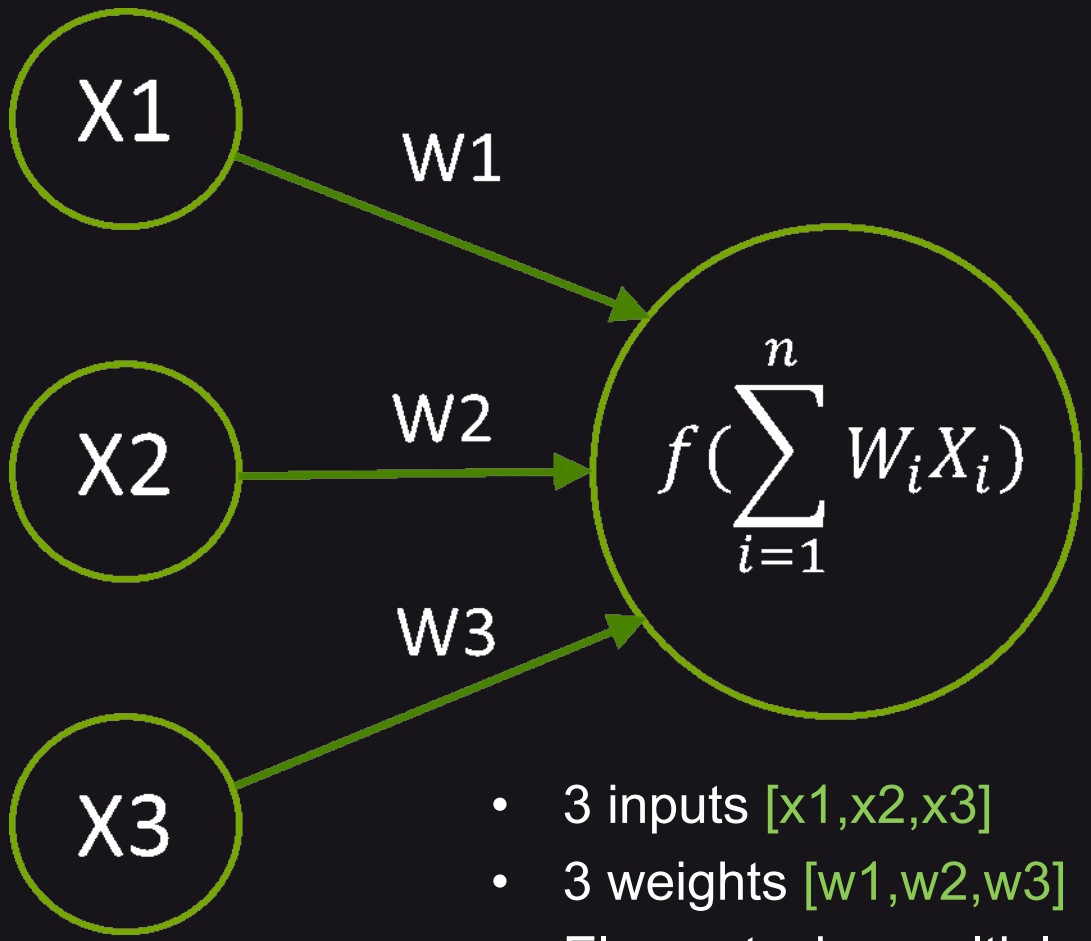
Jeremy Howard & Rachel Thomas
http://course.fast.ai

Andrej Karpathy
http://cs231n.github.io

François Chollet (Keras lead dev)
https://keras.io/

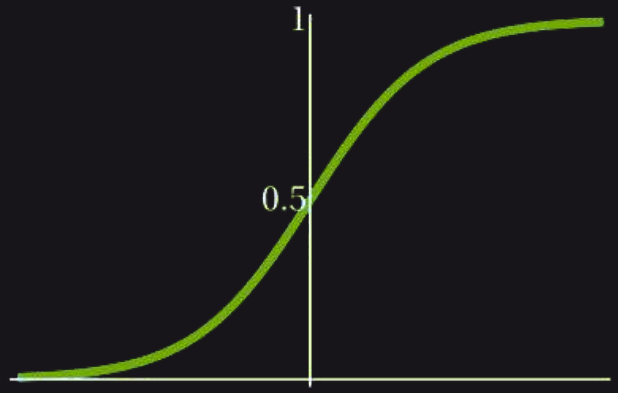
1.What is a neural network?

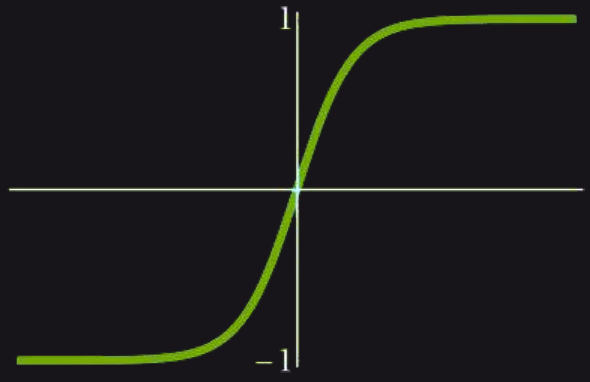
What is a neuron?

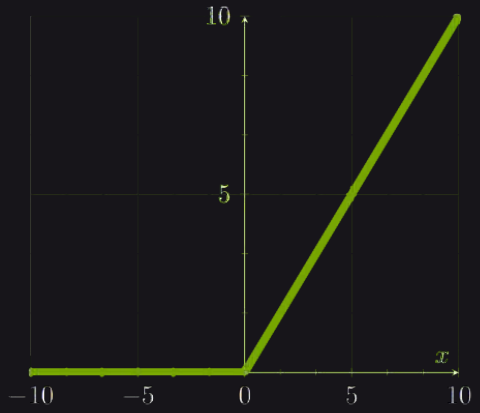


- 3 inputs [x1,x2,x3]
- 3 weights [w1,w2,w3]
- Element-wise multiply and sum
- Apply *activation function* **f**
- **Often add a bias too (weight of 1) – not shown**

What is an Activation Function?







sigmoid

$$f(x) = \frac{1}{1+e^{-(x)}}$$

tanh

$$\tanh(x) = \frac{e^{(x)} - e^{(-x)}}{e^{(x)} + e^{(x)}}$$

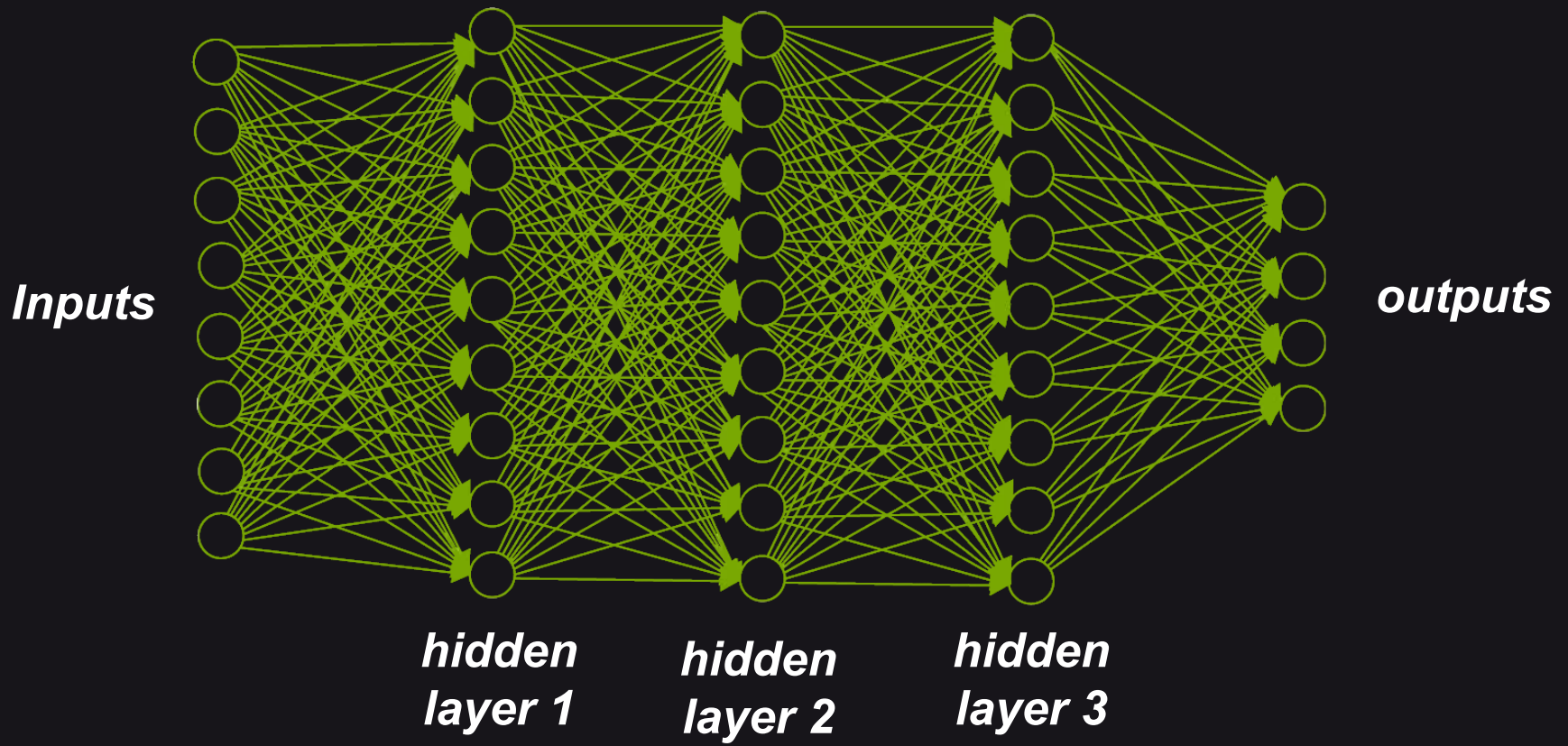
reLU

$$relu(x) = max(0, x)$$

Nonlinearities … "squashing functions" … transform neuron's output

NB: sigmoid output in [0,1]

What is a (Deep) Neural Network?



Outputs of one layer are inputs into the next layer

How does a neural network learn?

- We need labelled examples "training data"
- We initialize network weights randomly and initially get random predictions
- For each labelled training data point, we calculate the error between the network's predictions and the ground-truth labels
- Use 'backpropagation' (chain rule), to update the network parameters (weights + convolutional filters) in the opposite direction to the error

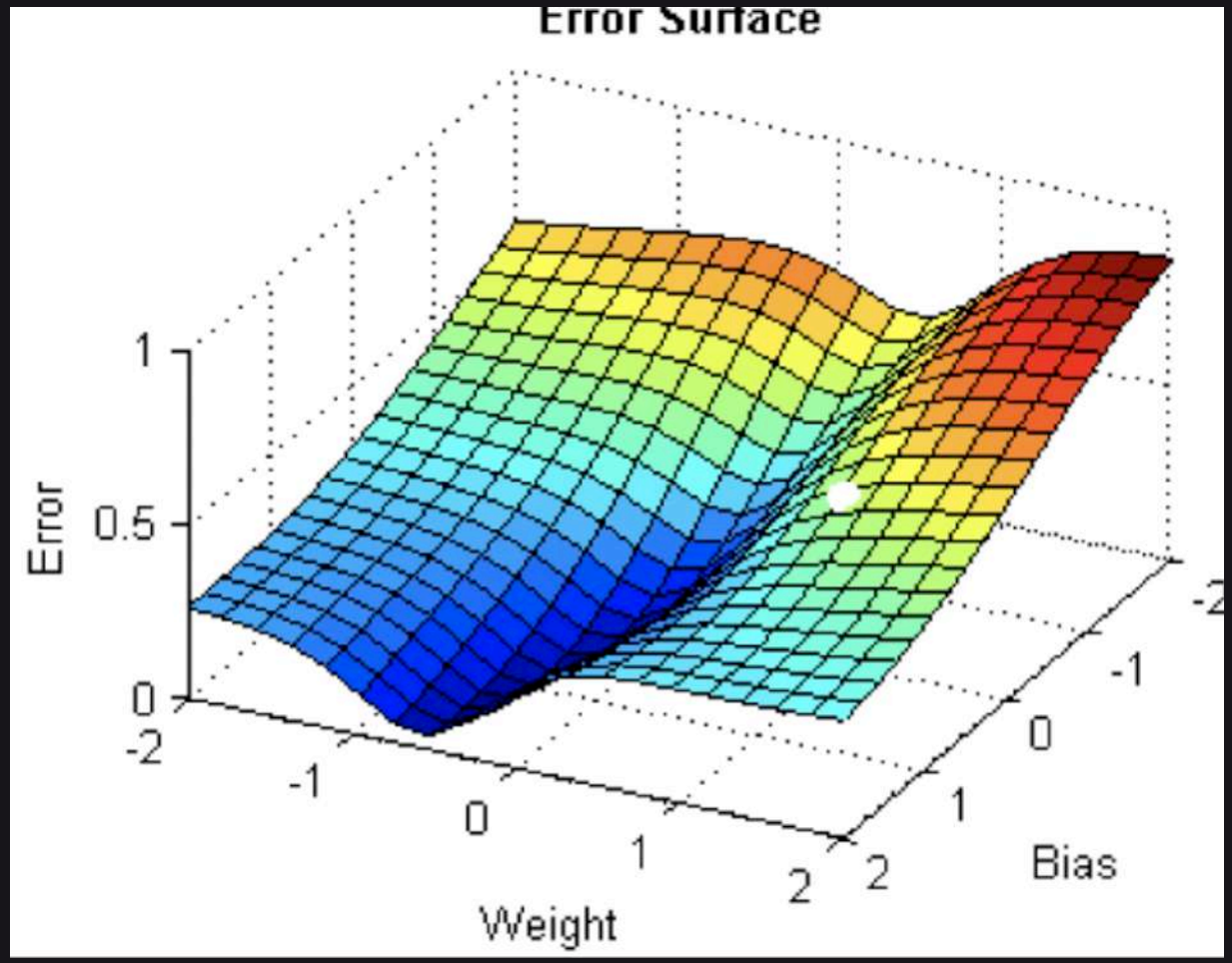
$$w_i = w^0{}_i - \eta \frac{\partial}{\partial w_i}(E)$$

How does a neural network learn?

$$w_i = w^0{}_i - \eta \frac{\partial}{\partial w_i}(E)$$

New weight = Old weight - (Learning rate x Gradient of weight with respect to Error)

Gradient Descent Interpretation



http://scs.ryerson.ca/~aharley/neural-networks/

http://playground.tensorflow.org

What is a Neural Network?

For much more detail, see:

1. Michael Nielson's Neural Networks & Deep Learning free online book

http://neuralnetworksanddeeplearning.com/chap1.html

2. Anrej Karpathy's CS231n Notes

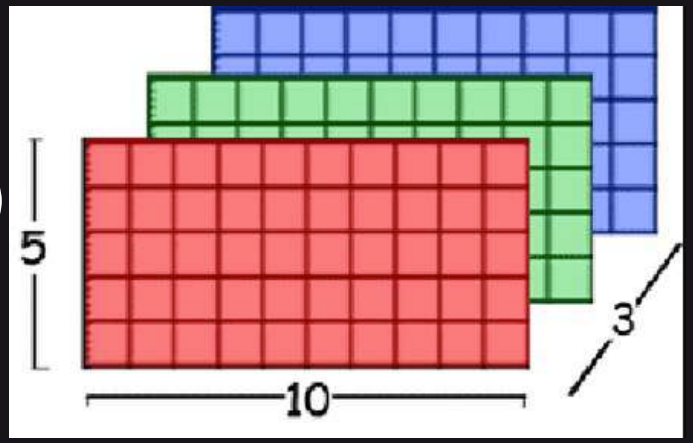
http://neuralnetworksanddeeplearning.com/chap1.html

2. What is a convolutional neural network?

What is a Convolutional Neural Network?

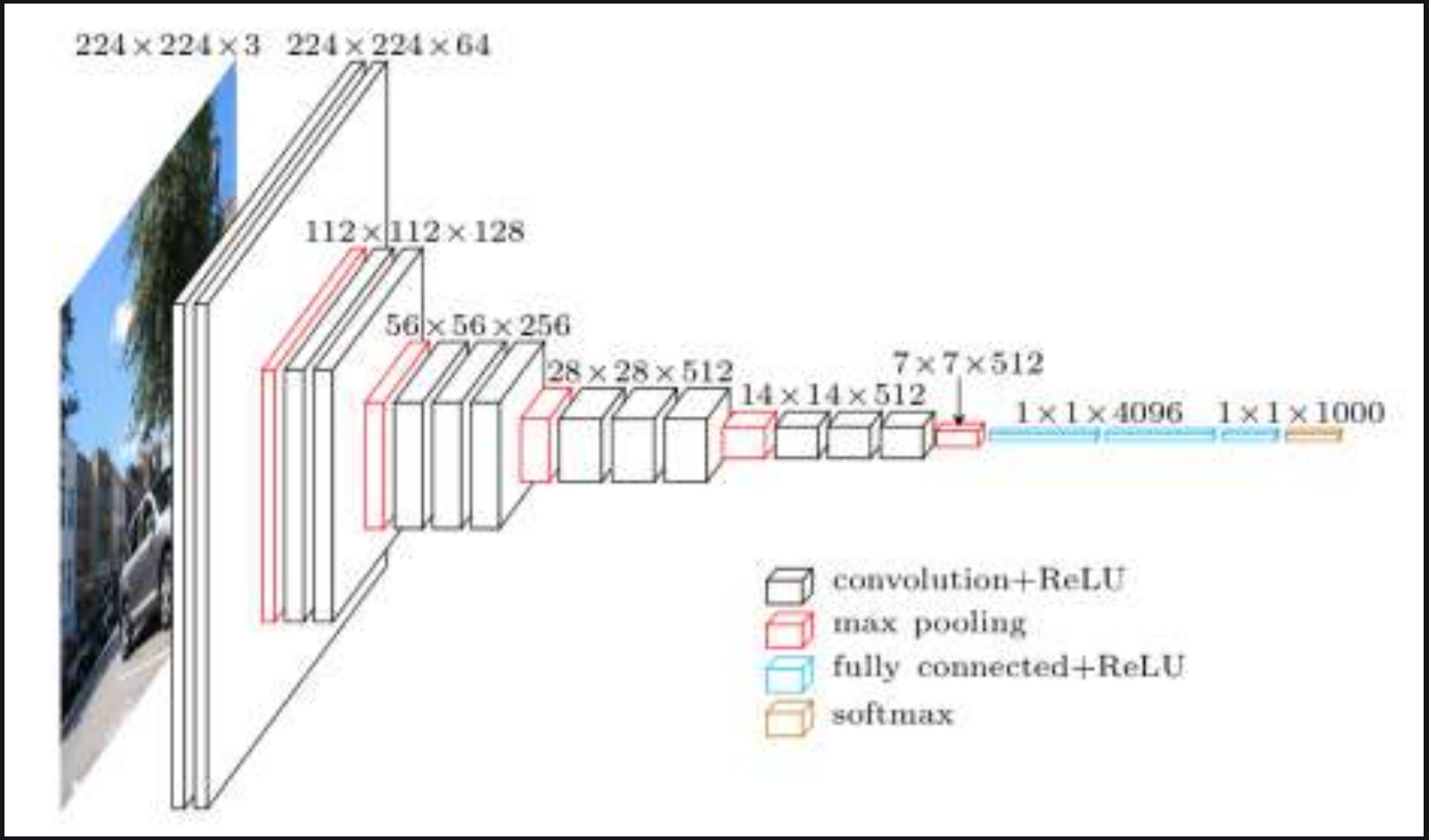
"like a ordinary neural network but with special types of layers that work well on images"

(math works on numbers)



- **Pixel = 3 colour channels (R, G, B)**
- **Pixel intensity ∈[0,255]**
- **Image has width w and height h**
- **Therefore image is w x h x 3 *numbers***

Example Architecture

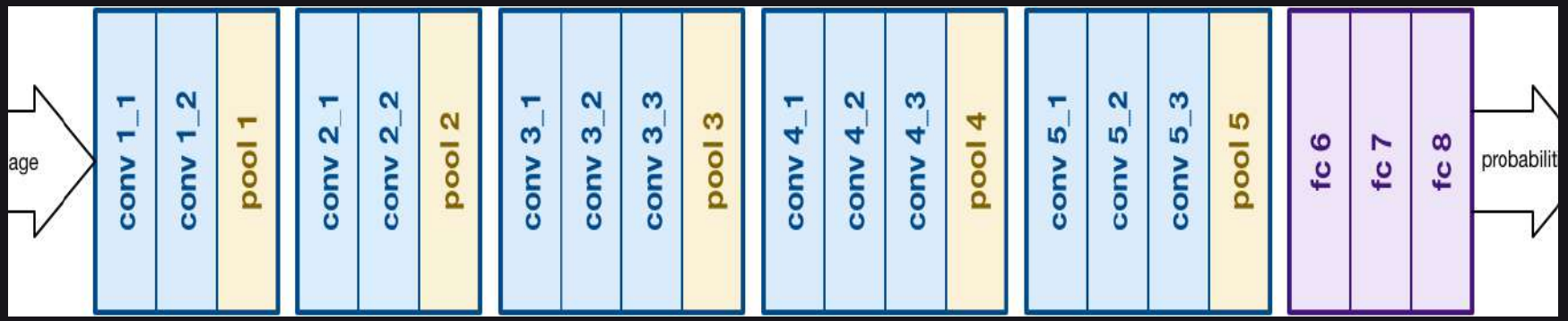


This is VGGNet – don't panic, we'll break it down piece by piece

Example Architecture

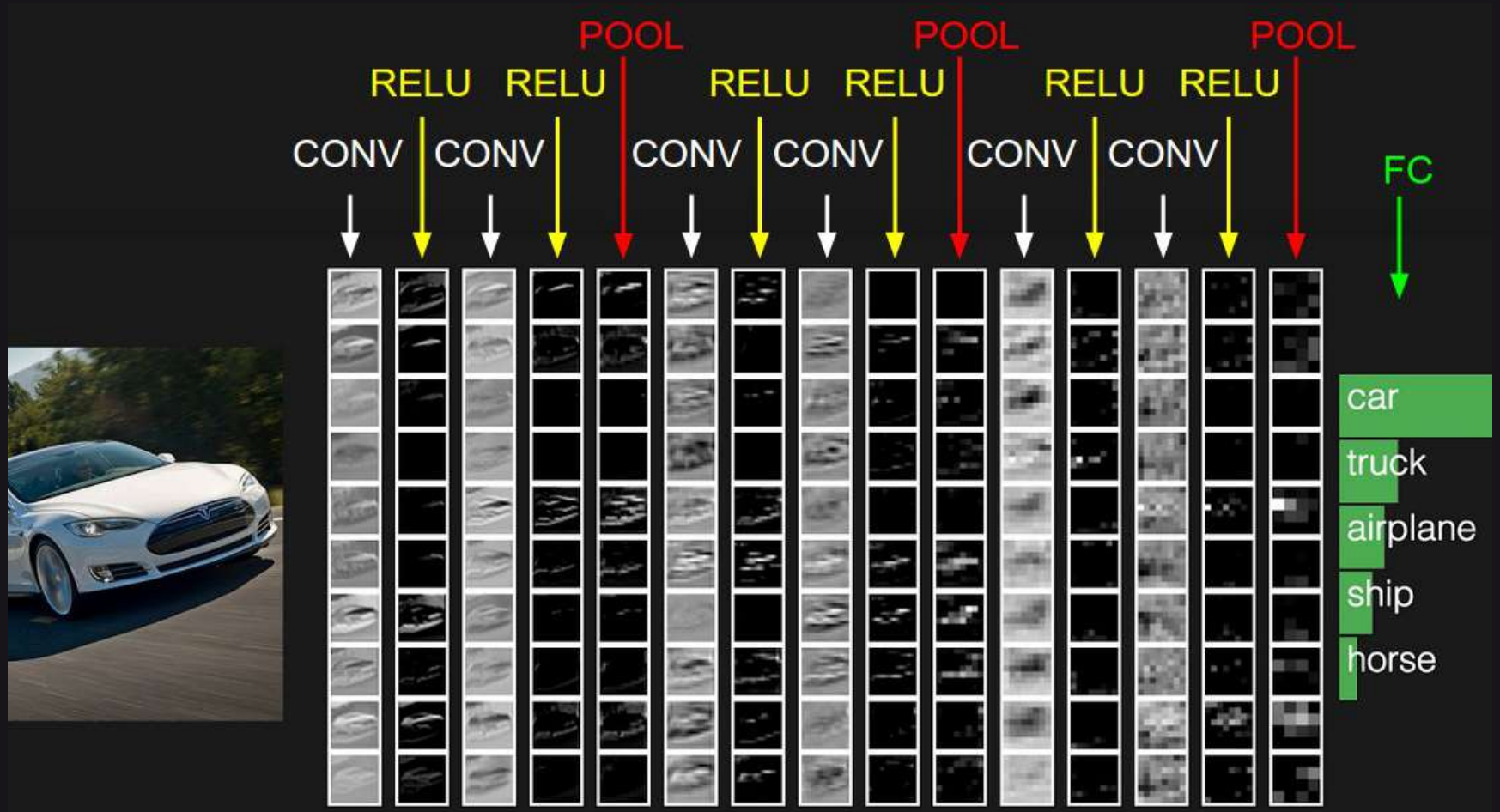
This is VGGNet – don't panic, we'll break it down piece by piece

VGGNet



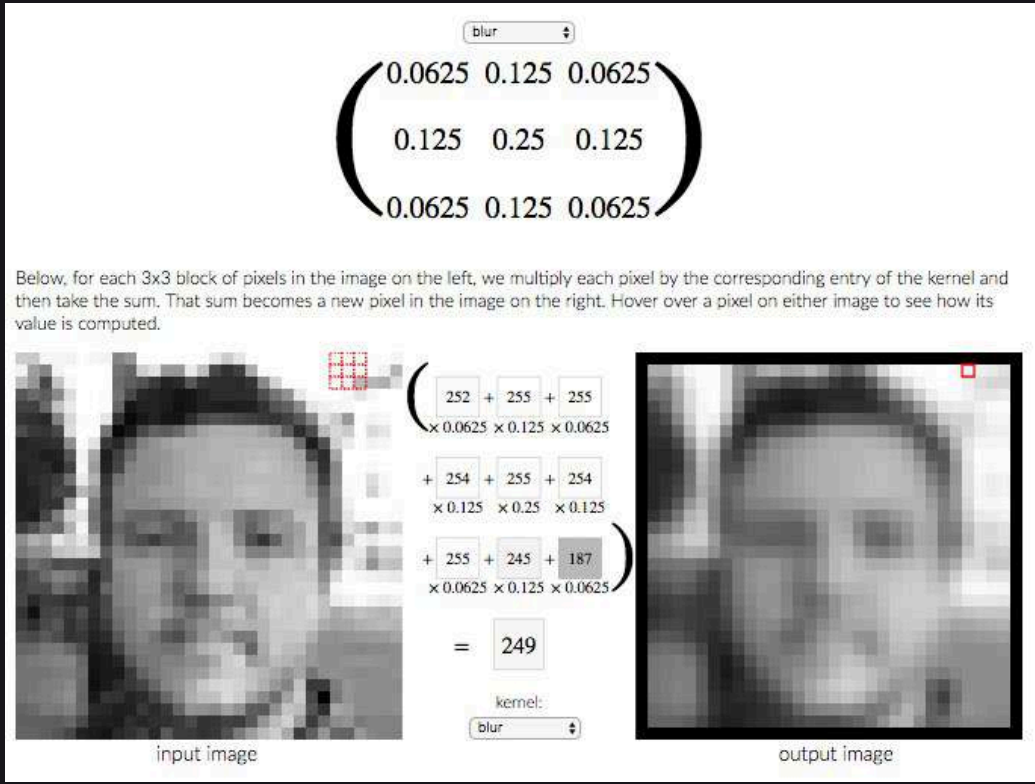
http://cs231n.github.io/convolutional-networks/

Alternating convolutions, activations, and pooling



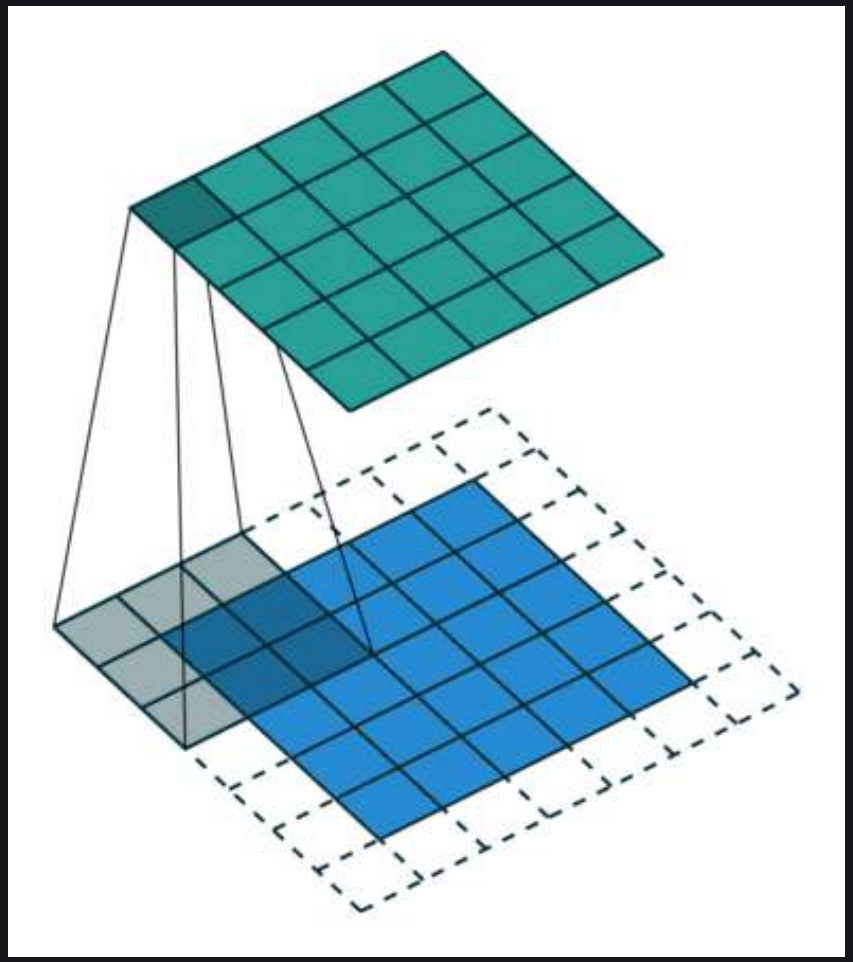
http://cs231n.github.io/convolutional-networks/

Convolutions



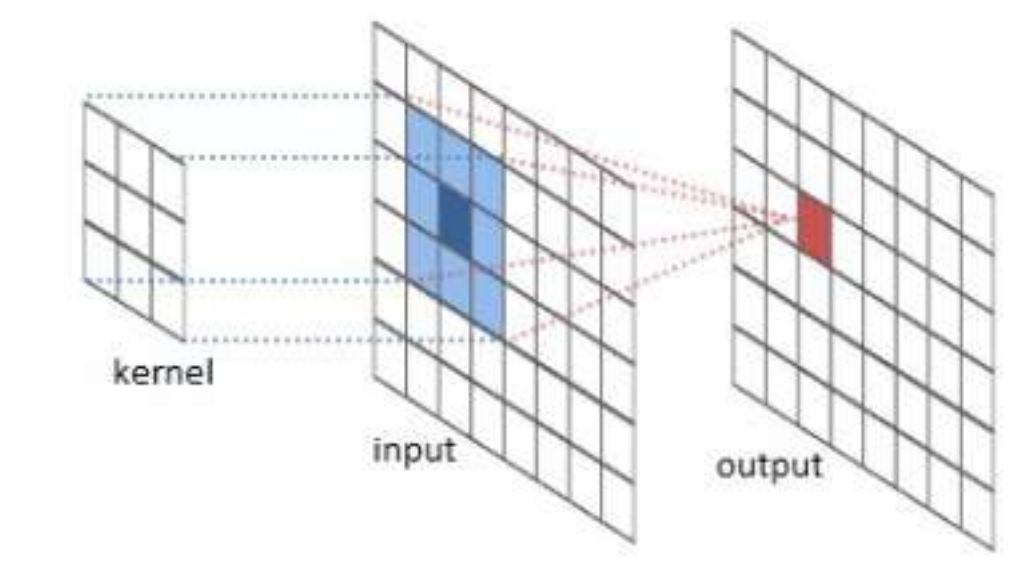
http://setosa.io/ev/image-kernels/

Convolutions



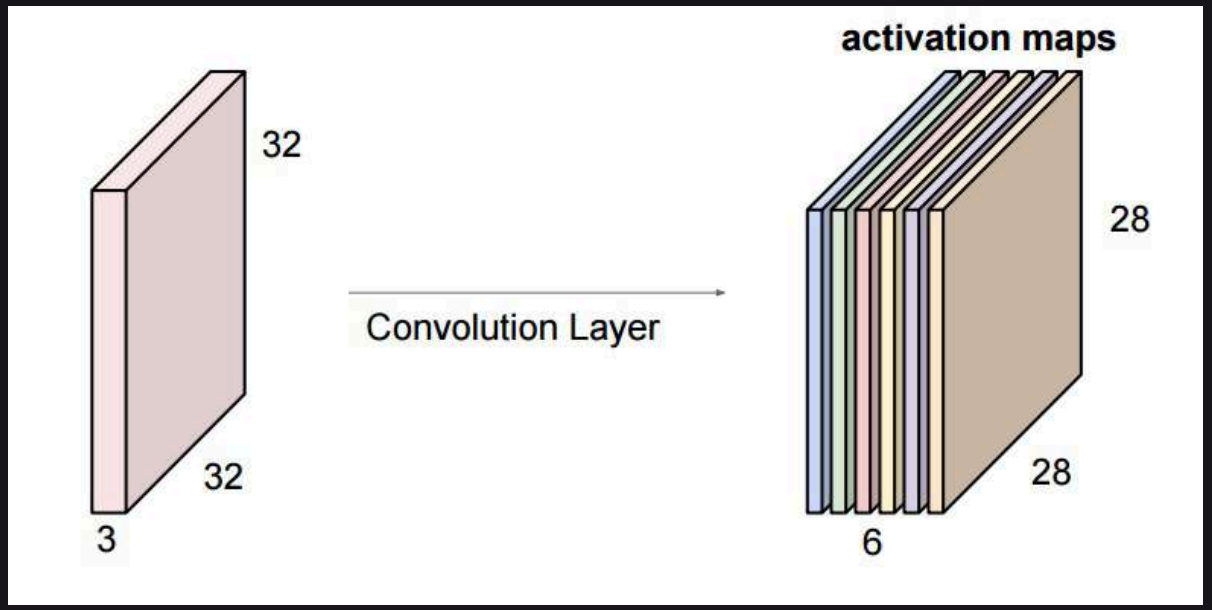
http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

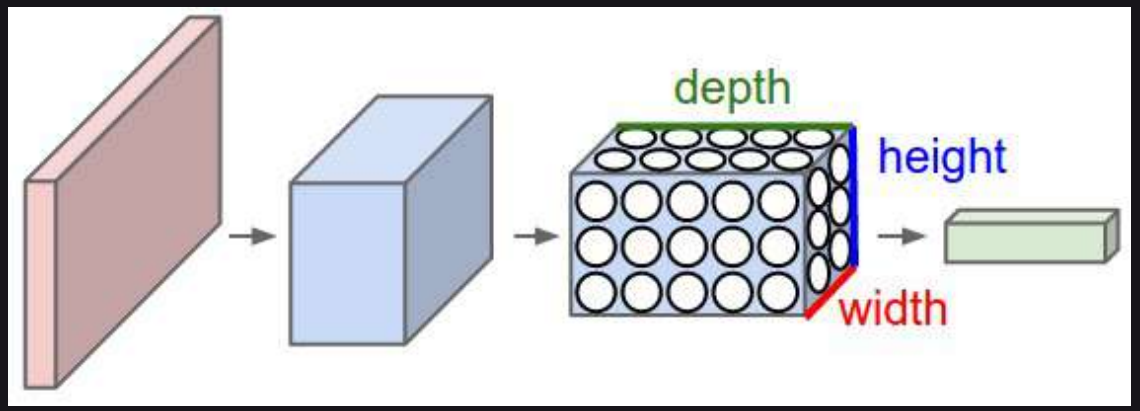
New Layer Type: Convolutional Layer



- 2-d weighted average when multiply kernel over pixel patches
- We slide the kernel over all pixels of the image (handle borders)
- Kernel starts off with “random” values and network updates *(learns)* the kernel values (using *backpropagation*) to try minimize loss
- Kernels shared across the whole image (parameter sharing)

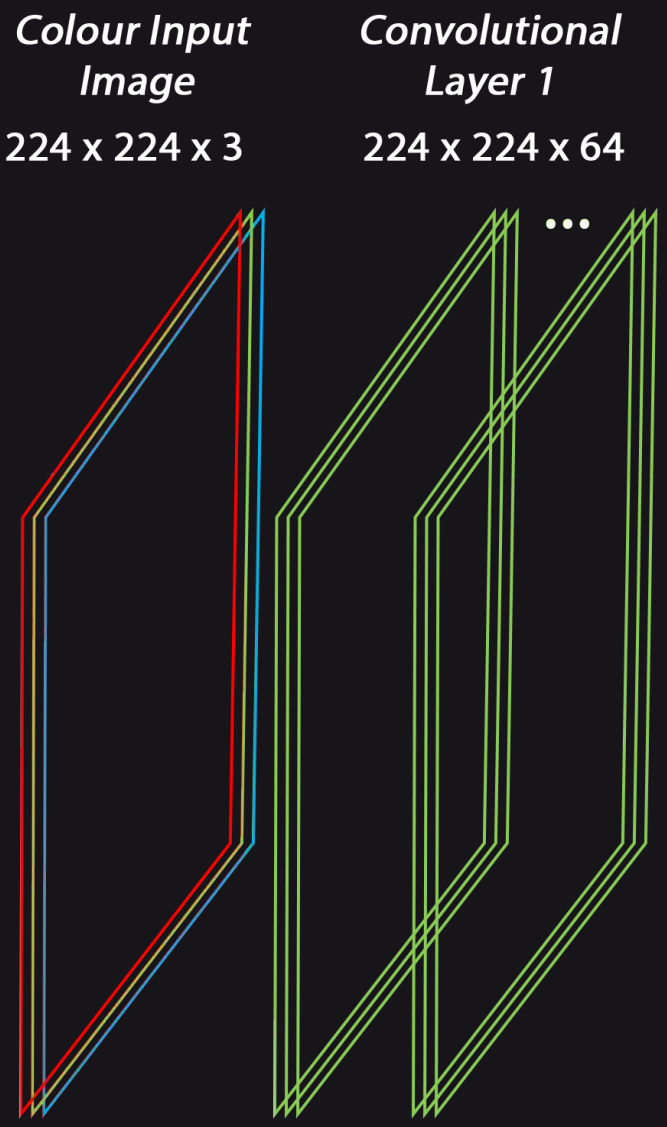
Many Kernels = Many "Activation Maps" = Volume





http://cs231n.github.io/convolutional-networks/

New Layer Type: Convolutional Layer



Convolutions

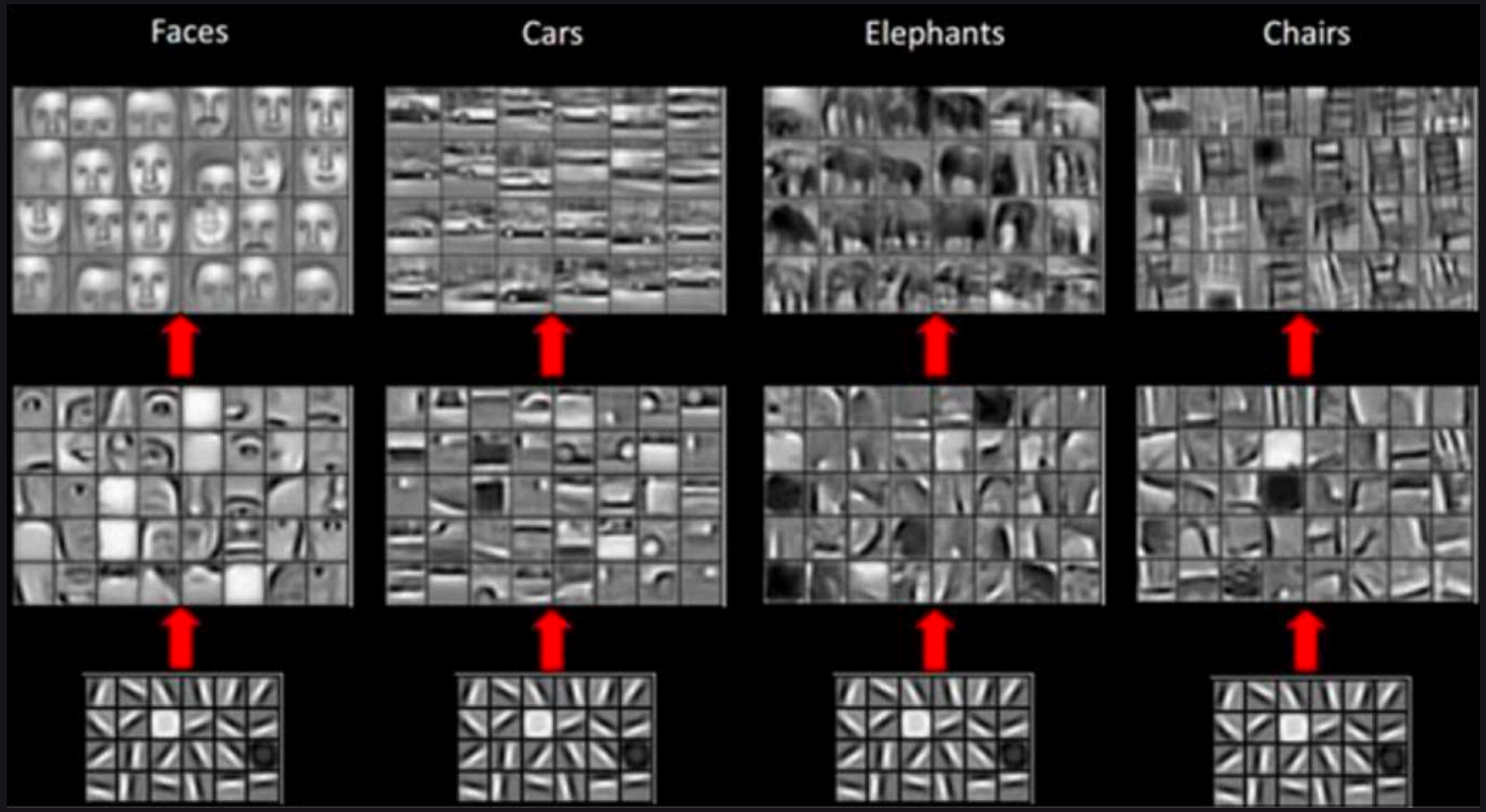
Visualizing and Understanding Convolutional Networks

Matthew D. Zeiler ZEILER@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University

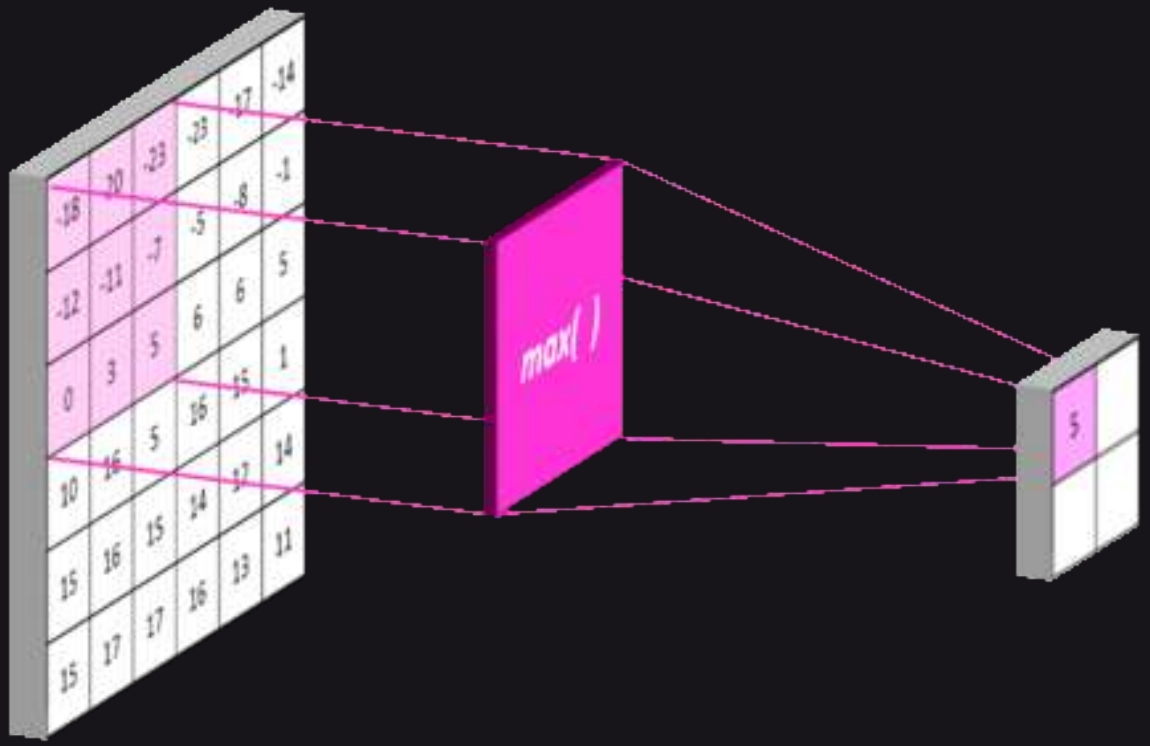
Rob Fergus FERGUS@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University

https://**github.com/fchollet/keras/blob/master/examples/conv_filter_visualization.py**

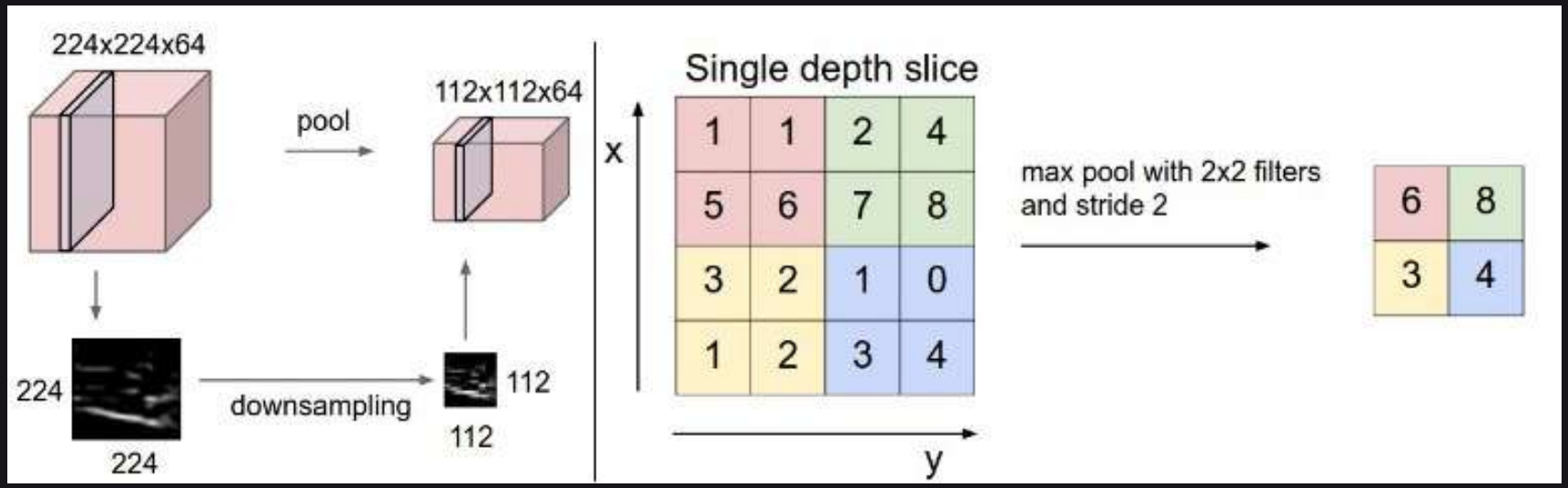
Convolution Learn Hierarchical Features



New Layer Type: Max Pooling



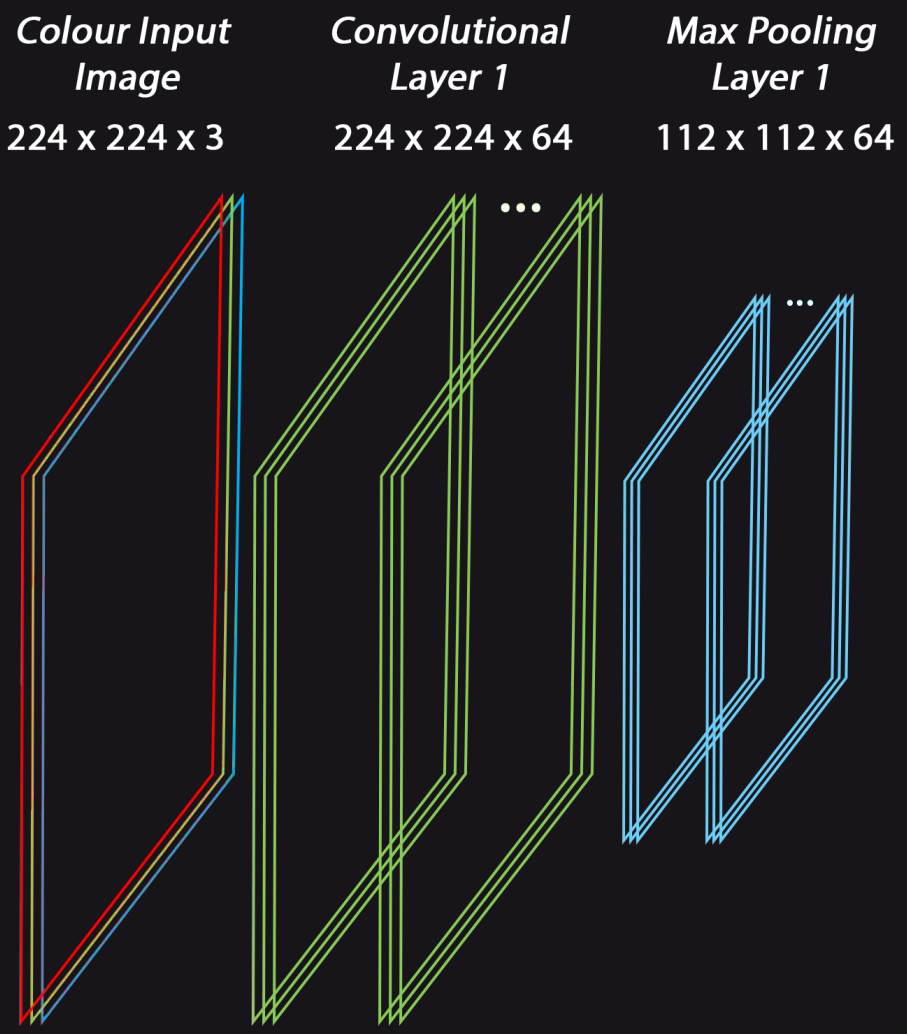
New Layer Type: Max Pooling



- Reduces dimensionality from one layer to next
- …by replacing NxN sub-area with max value
- Makes network "look" at larger areas of the image at a time
- e.g. Instead of identifying fur, identify cat
- Reduces overfitting since losing information helps the network generalize

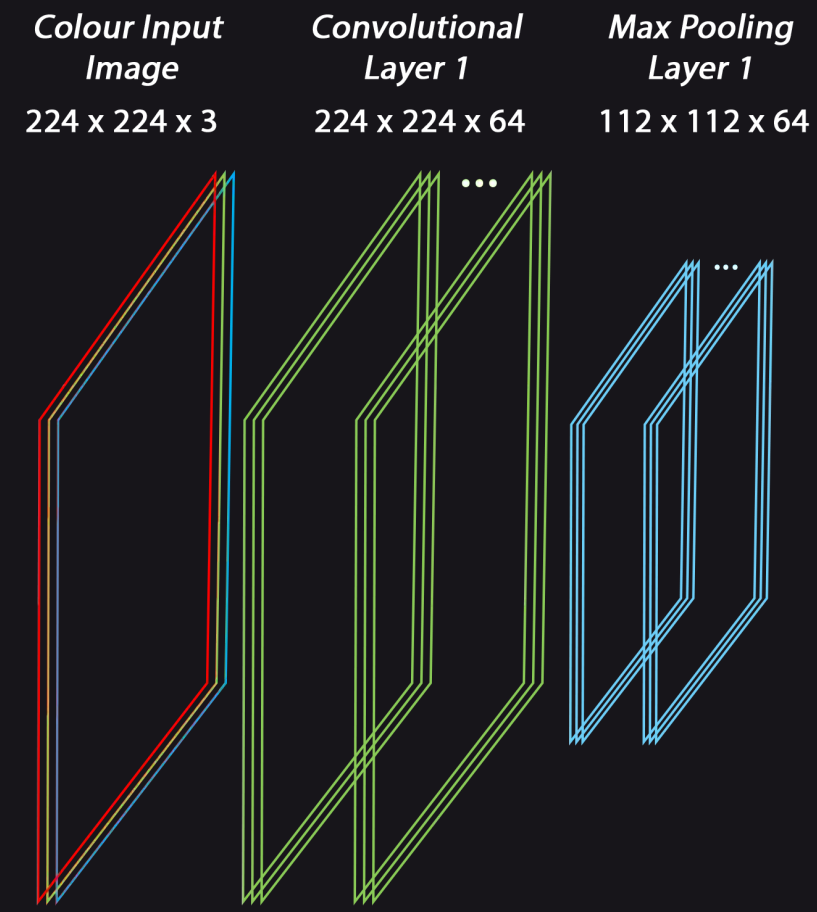
http://cs231n.github.io/convolutional-networks/

New Layer Type: Max Pooling



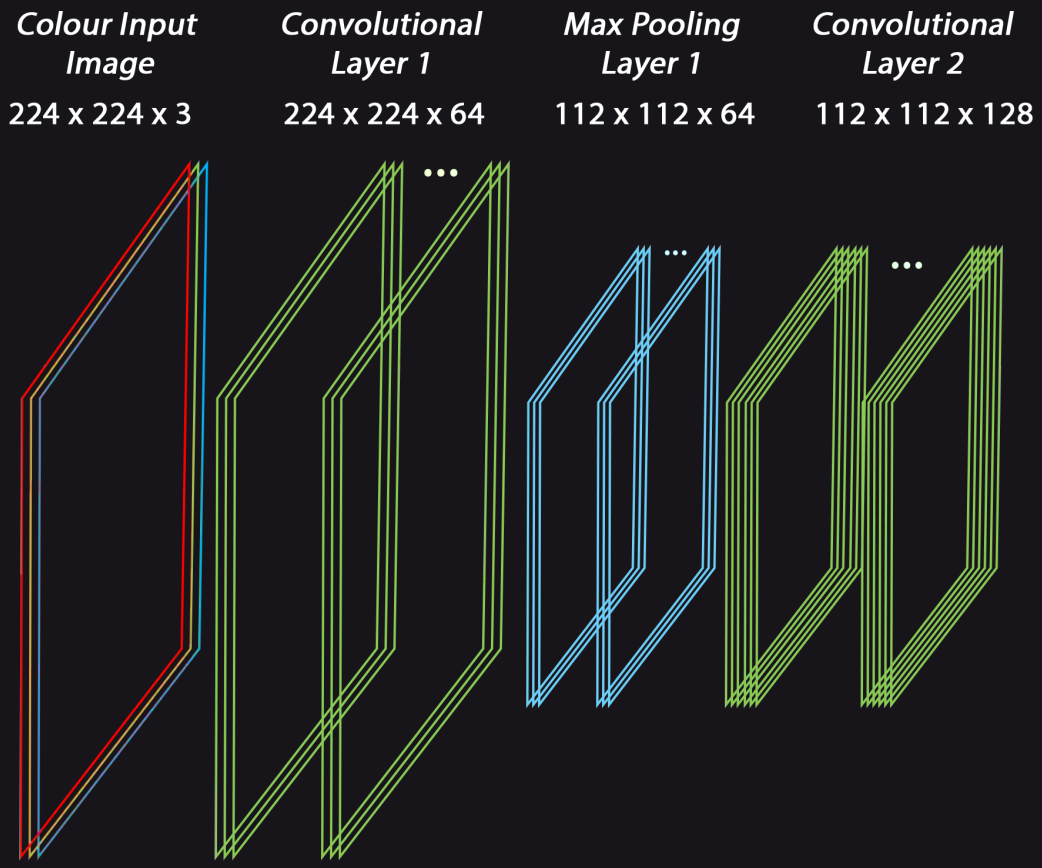
Stack Conv + Pooling Layers and Go Deep

Convolution + max pooling



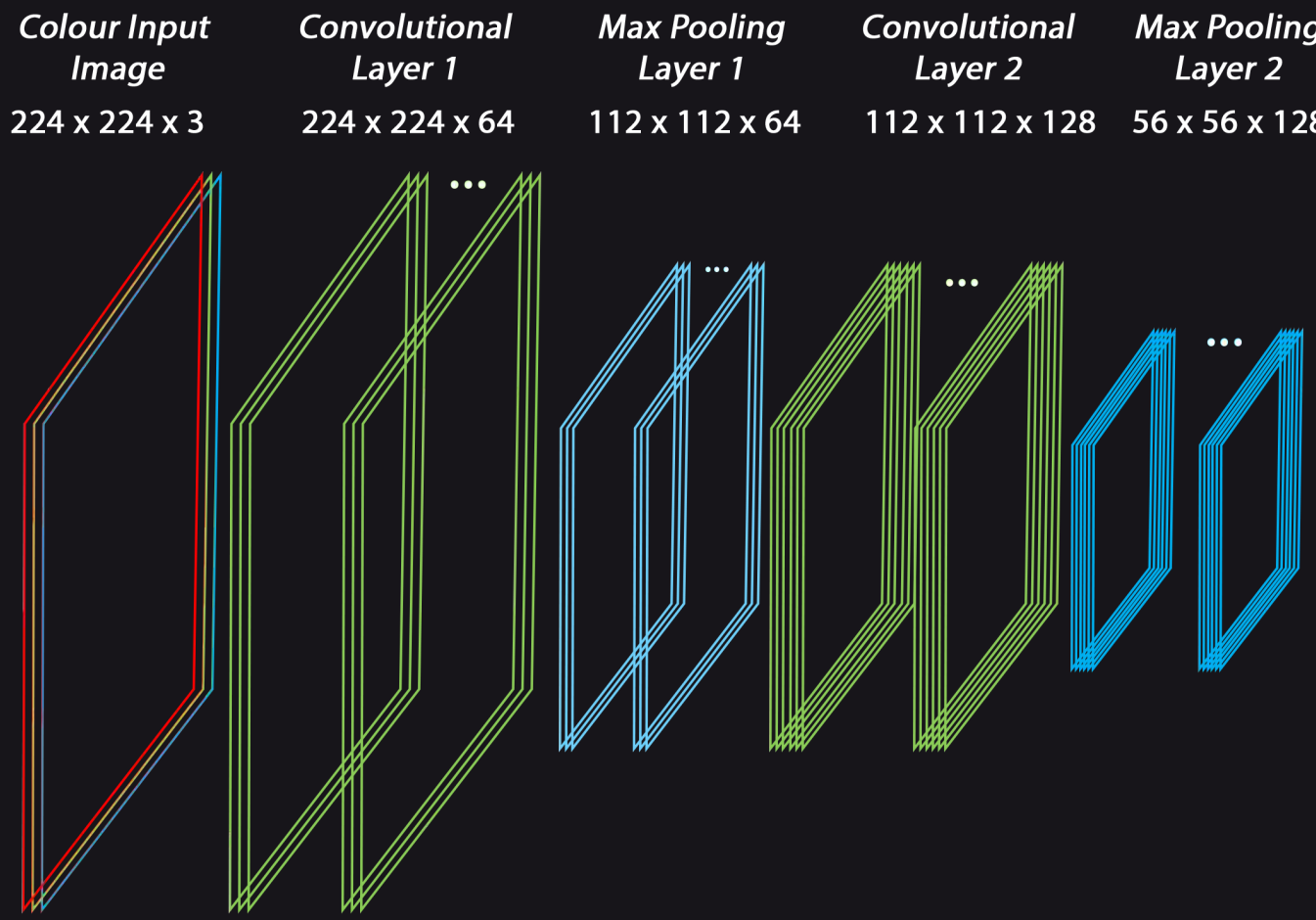
Stack Conv + Pooling Layers and Go Deep

Convolution + max pooling



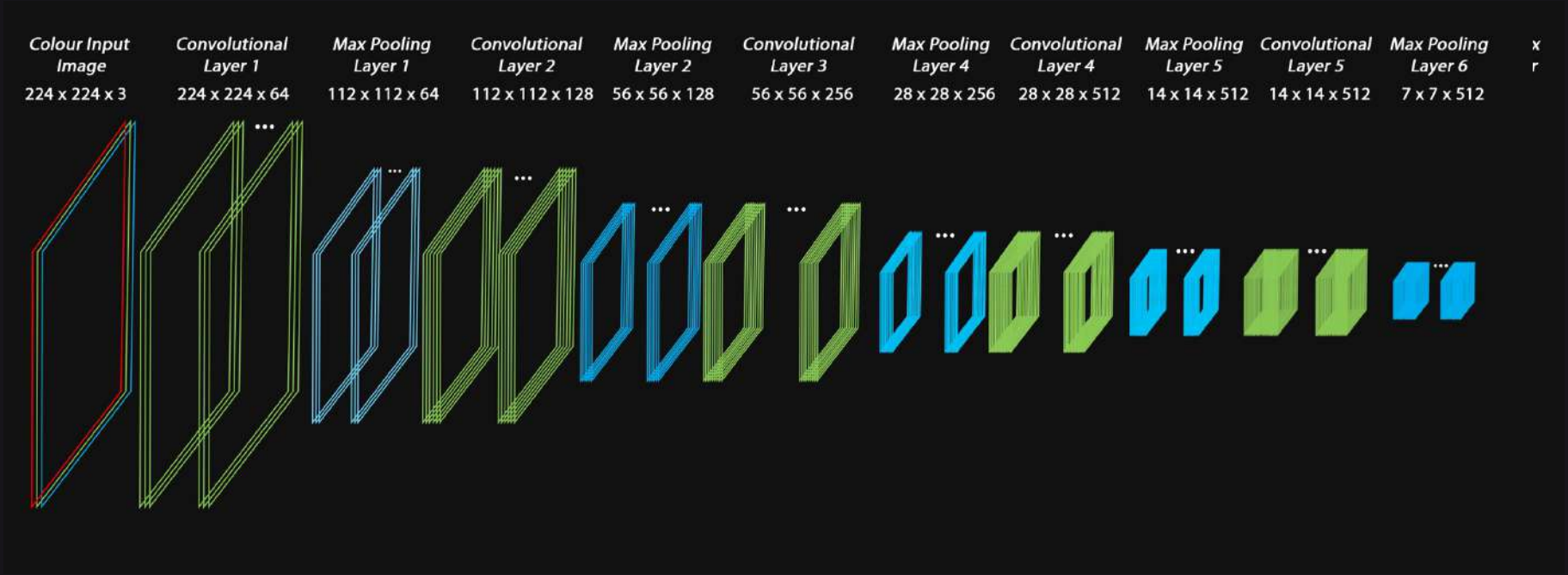
Stack These Layers and Go Deep

Convolution + max pooling



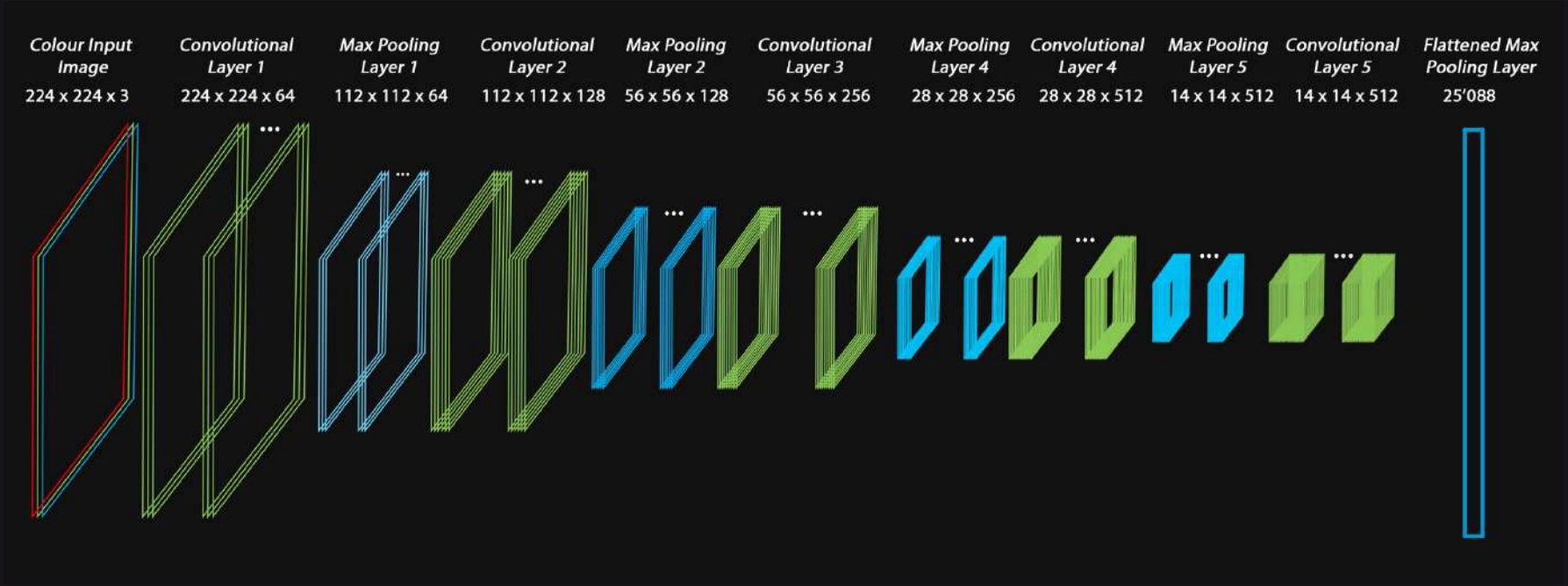
Stack These Layers and Go Deep

Convolution + max pooling



Flatten the Final “Bottleneck” layer

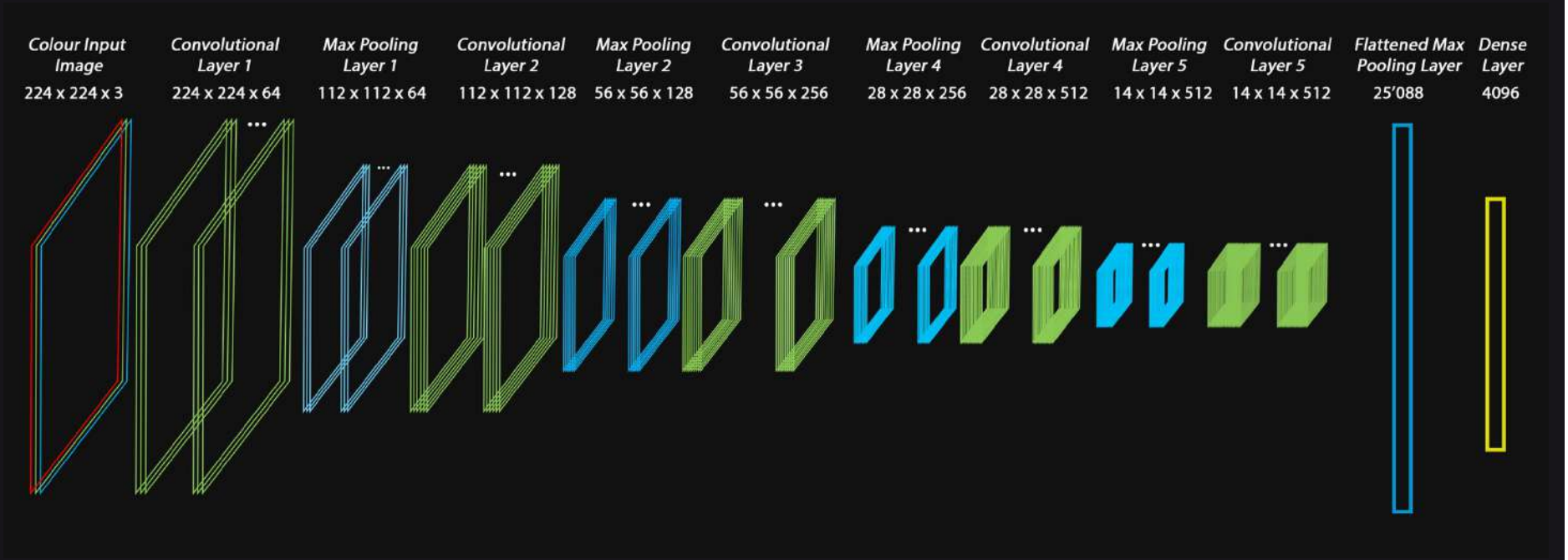
Convolution + max pooling



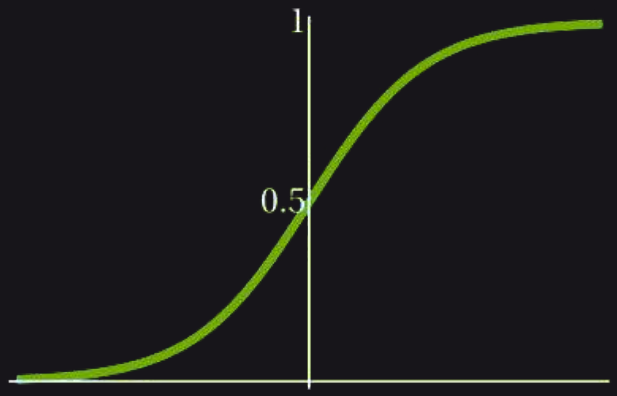
Flatten the final 7 x 7 x 512 max pooling layer
Add fully-connected dense layer on top

Bringing it all together

Convolution + max pooling



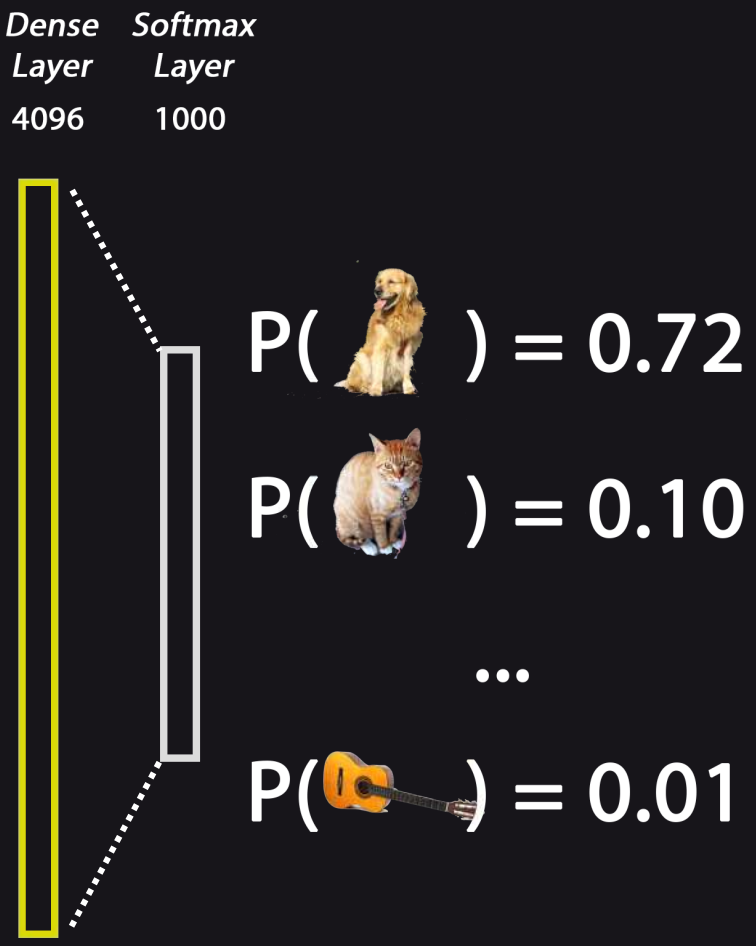
Softmax



Convert scores $\in \mathbb{R}$ to probabilities $\in [0,1]$

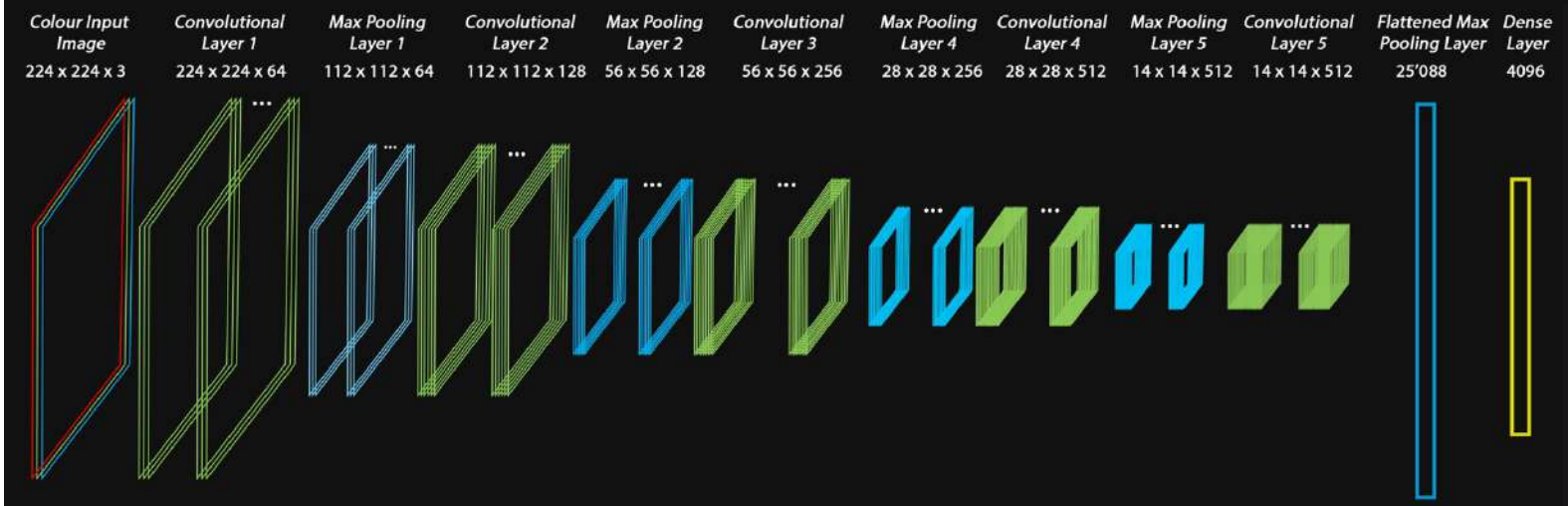
Final output prediction = highest probability class

$$\phi_{softmax}(z) = \frac{e_i^z}{\sum_{m=1}^{M} e_m^z}$$



Bringing it all together

Convolution + max pooling + fully connect



We need labelled training data!

ImageNet

1000 object categories

1.2 million training images

http://image-net.org/explore

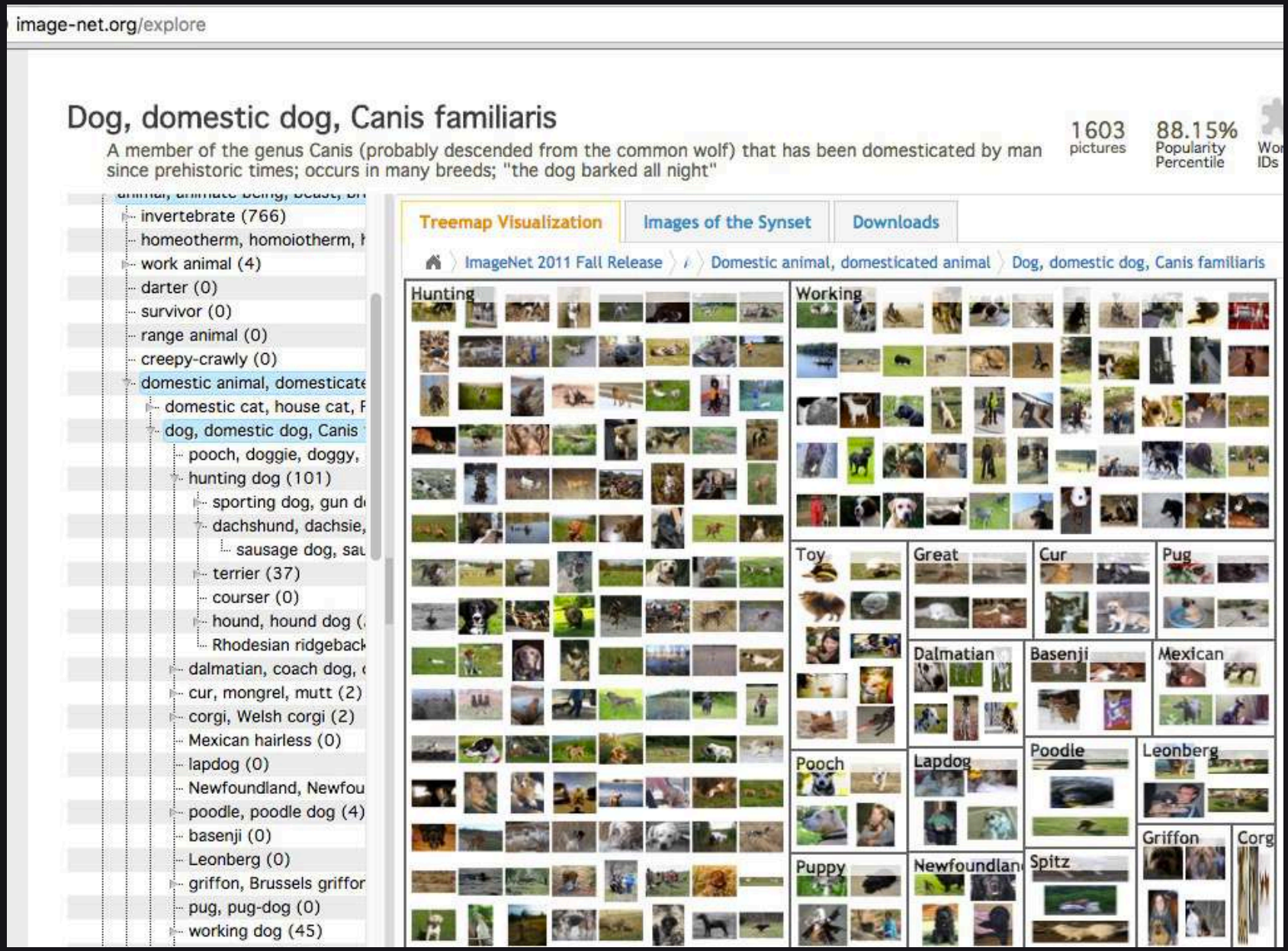


image-net.org/explore
Dog, domestic dog, Canis familiaris
A member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds; "the dog barked all night"
1603 pictures
88.15% Popularity Percentile
invertebrate (766)
homeotherm, homoiotherm,
work animal (4)
darter (0)
survivor (0)
range animal (0)
creepy-crawly (0)
domestic animal, domesticate
domestic cat, house cat,
dog, domestic dog, Canis
pooch, doggie, doggy,
hunting dog (101)
sporting dog, gun d
dachshund, dachsie,
sausage dog, sau
terrier (37)
courser (0)
hound, hound dog (
Rhodesian ridgeback
dalmatian, coach dog,
cur, mongrel, mutt (2)
corgi, Welsh corgi (2)
Mexican hairless (0)
lapdog (0)
Newfoundland, Newfou
poodle, poodle dog (4)
basenji (0)
Leonberg (0)
griffon, Brussels griffor
pug, pug-dog (0)
working dog (45)
Treemap Visualization
Images of the Synset
Downloads
ImageNet 2011 Fall Release
Domestic animal, domesticated animal
Dog, domestic dog, Canis familiaris
Hunting
Working
Toy
Great
Cur
Pug
Dalmatian
Basenji
Mexican
Pooch
Lapdog
Poodle
Leonberg
Puppy
Newfoundlan
Spitz
Griffon
Corg

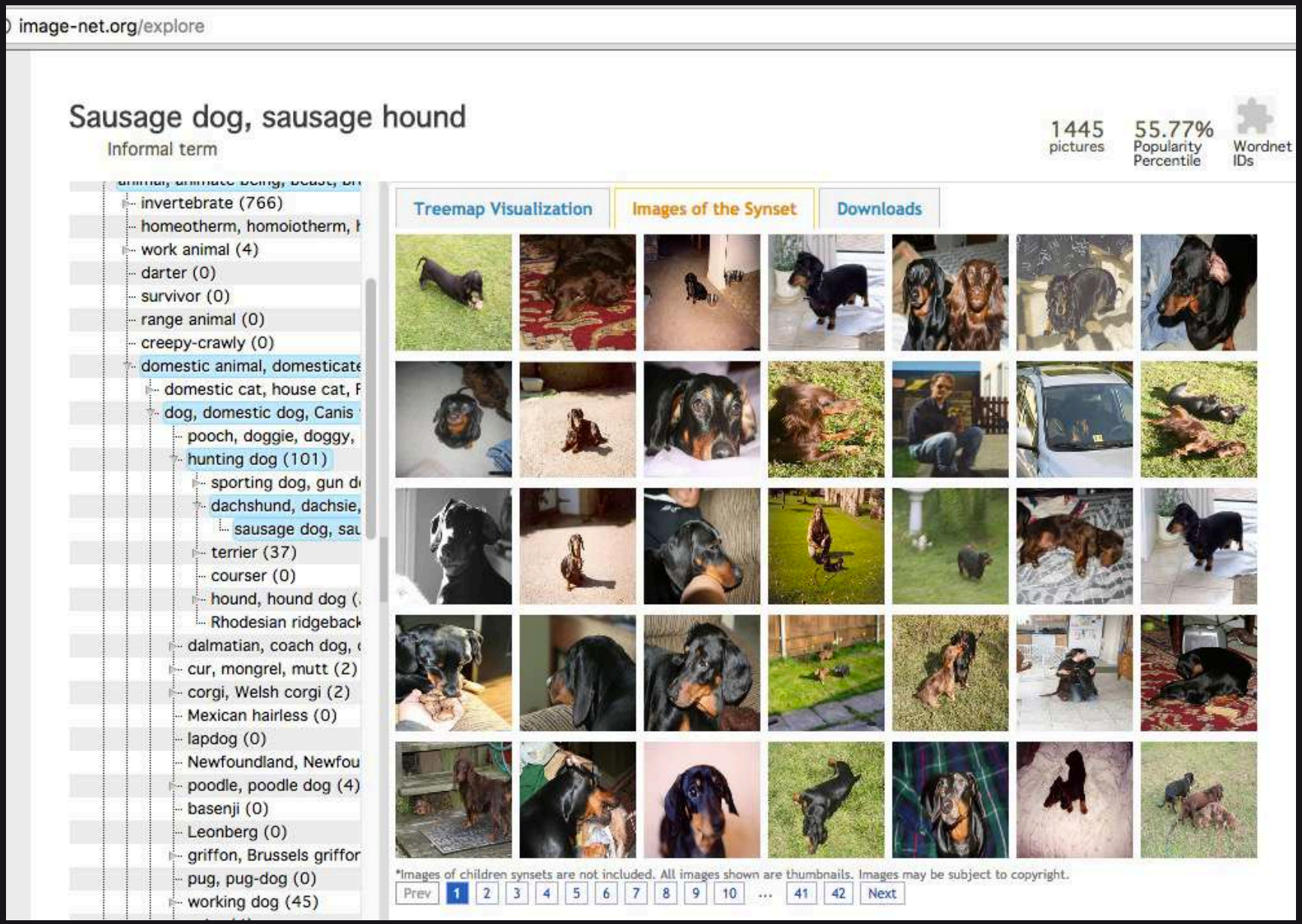
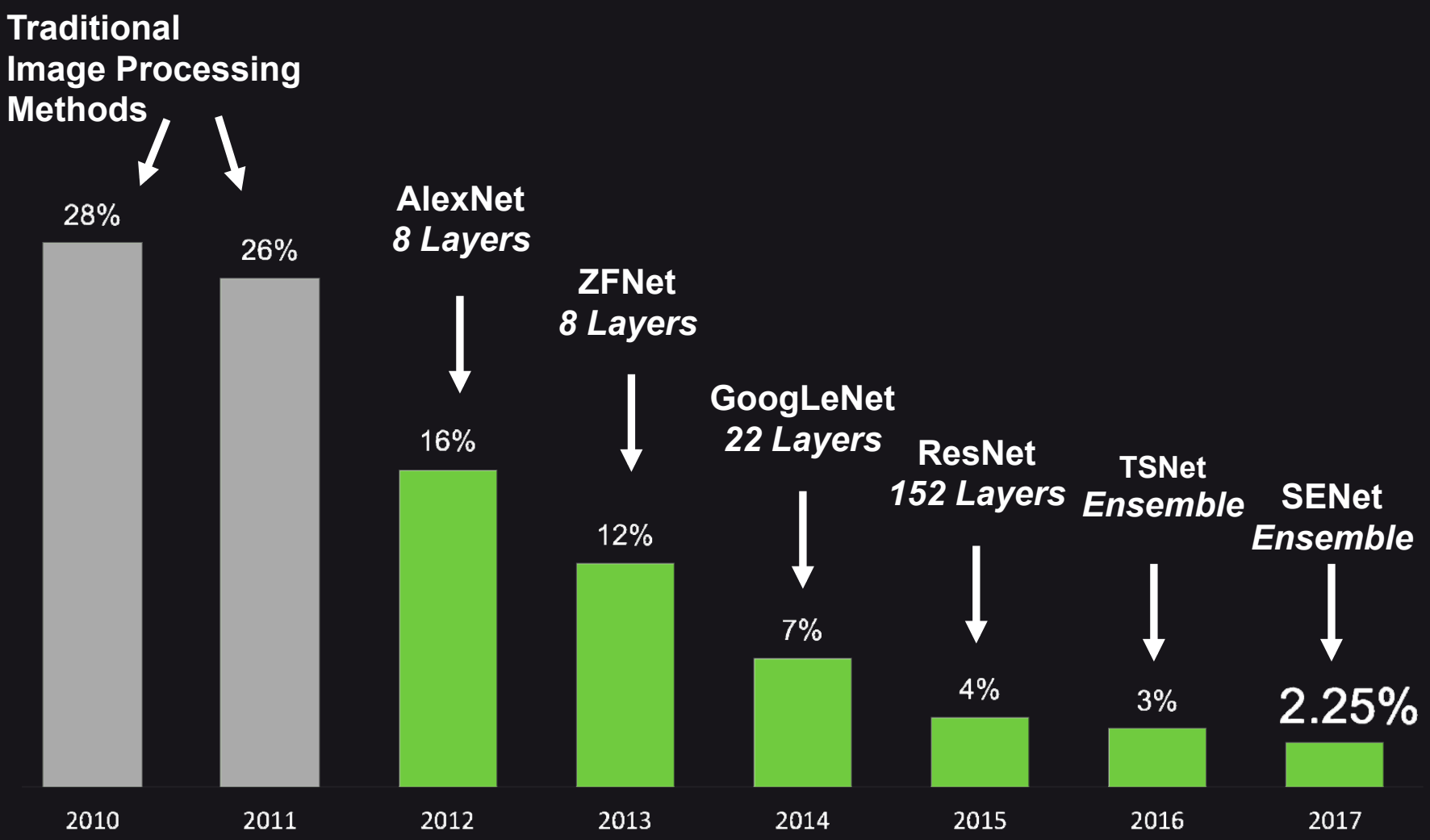


image-net.org/explore
Sausage dog, sausage hound
Informal term
1445
pictures
55.77%
Popularity
Percentile
Wordnet
IDs
invertebrate (766)
homeotherm, homoiotherm,
work animal (4)
darter (0)
survivor (0)
range animal (0)
creepy-crawly (0)
domestic animal, domesticate
domestic cat, house cat,
dog, domestic dog, Canis
pooch, doggie, doggy,
hunting dog (101)
sporting dog, gun d
dachshund, dachsie,
sausage dog, sau
terrier (37)
courser (0)
hound, hound dog (
Rhodesian ridgeback
dalmatian, coach dog,
cur, mongrel, mutt (2)
corgi, Welsh corgi (2)
Mexican hairless (0)
lapdog (0)
Newfoundland, Newfou
poodle, poodle dog (4)
basenji (0)
Leonberg (0)
griffon, Brussels griffor
pug, pug-dog (0)
working dog (45)
Treemap Visualization
Images of the Synset
Downloads
*Images of children synsets are not included. All images shown are thumbnails. Images may be subject to copyright.
Prev
1
2
3
4
5
6
7
8
9
10
...
41
42
Next

ImageNet Top 5 Error Rate



3. How to use a convolutional neural network

Using a Pre-Trained ImageNet-Winning CNN

- **We've been looking at "VGGNet"**
- **Oxford Visual Geometry Group (VGG)**
- **ImageNet 2014 Runner-up**
- **Network is 16 layers (deep!)**
- **Easy to fine-tune**

https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

Example: Classifying Product Images

Classifying products into 9 categories

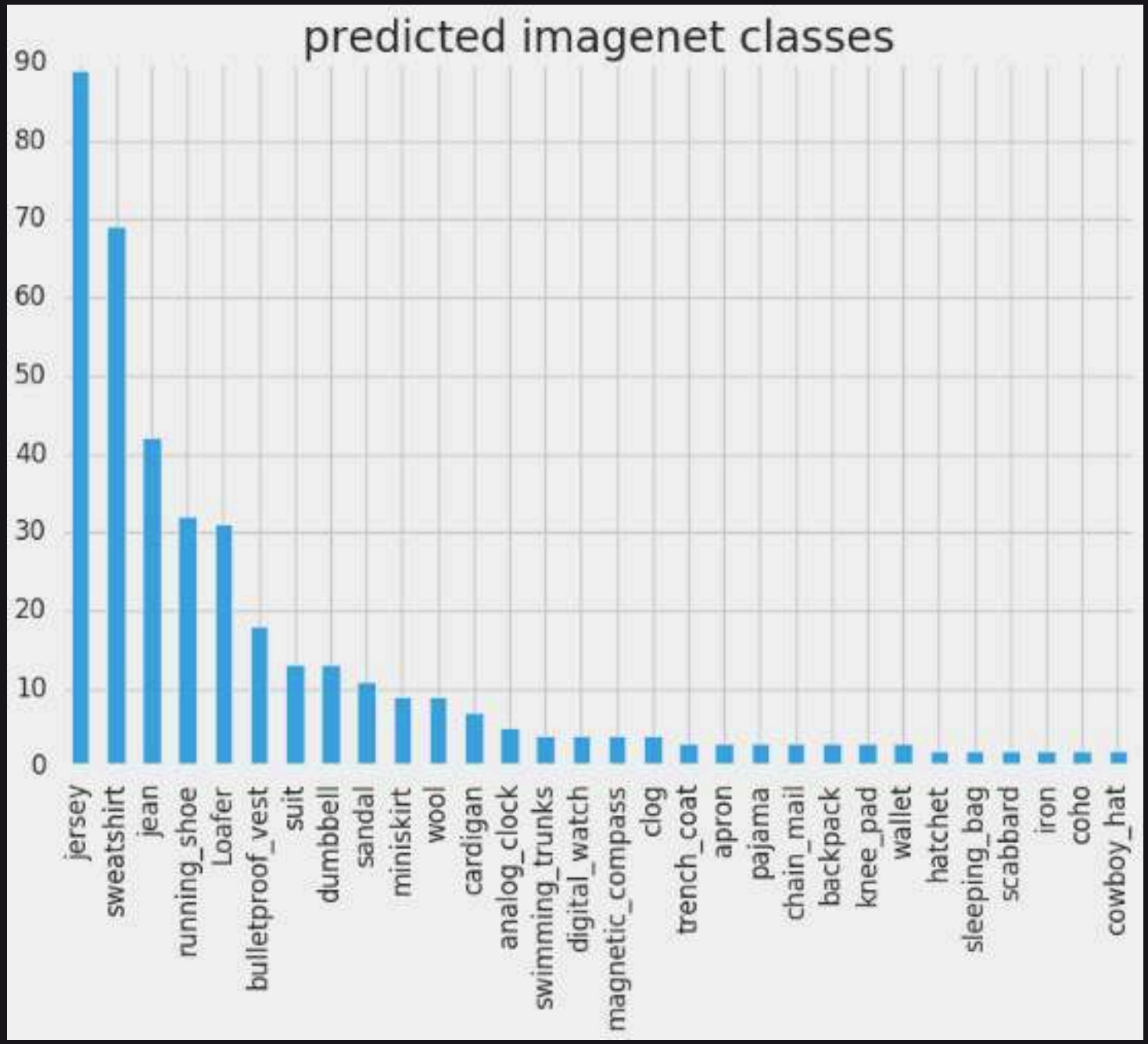
https://github.com/alexcnwy/CTDL_CNN_TALK_20170620

Start with Pre-Trained ImageNet Model

```
1 from keras.applications.vgg16 import VGG16
2 from keras.preprocessing import image
3 from keras.applications.vgg16 import preprocess_input
4 import numpy as np
5
6 model = VGG16(weights='imagenet', include_top=False)
7
8 img_path = 'elephant.jpg'
9 img = image.load_img(img_path, target_size=(224, 224))
10 x = image.img_to_array(img)
11 x = np.expand_dims(x, axis=0)
12 x = preprocess_input(x)
13
14 preds = model.predict(x)
15 print('Predicted:', decode_predictions(preds, top=3)[0])
16 # Predicted: [(u'n02504013', u'Indian_elephant', 0.82658225), (u'n01871265', u'tusker'
```

Consumers vs Producers of Machine Learning

https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

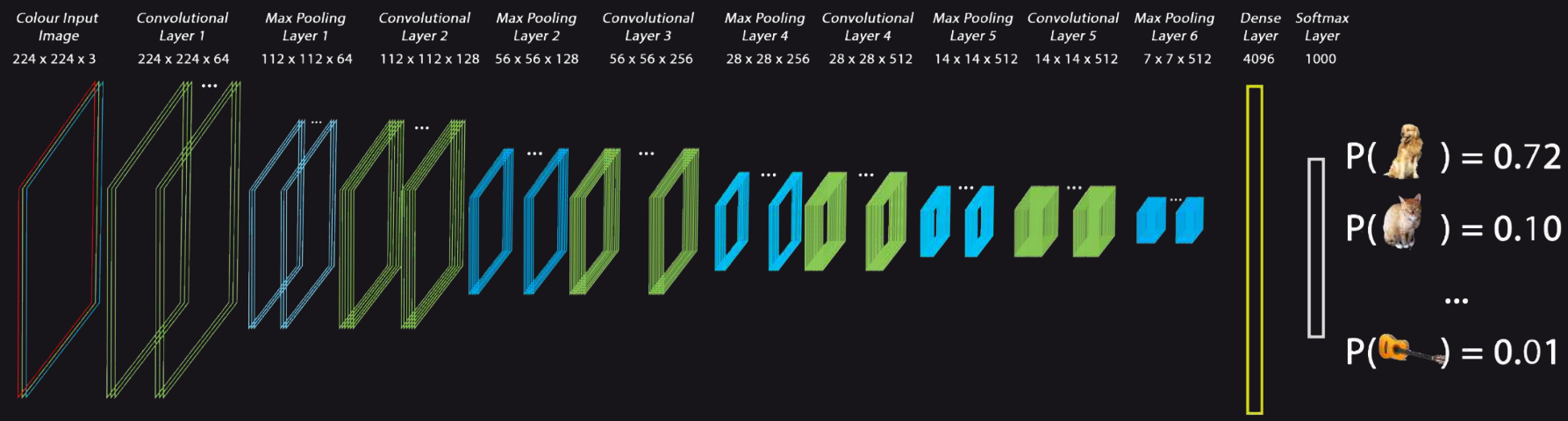

predicted imagenet classes
90
80
70
60
50
40
30
20
10
0
jersey
sweatshirt
jean
running_shoe
Loafer
bulletproof_vest
suit
dumbbell
sandal
miniskirt
wool
cardigan
analog_clock
swimming_trunks
digital_watch
magnetic_compass
clog
trench_coat
apron
pajama
chain_mail
backpack
knee_pad
wallet
hatchet
sleeping_bag
scabbard
iron
coho
cowboy_hat

“Transfer Learning”

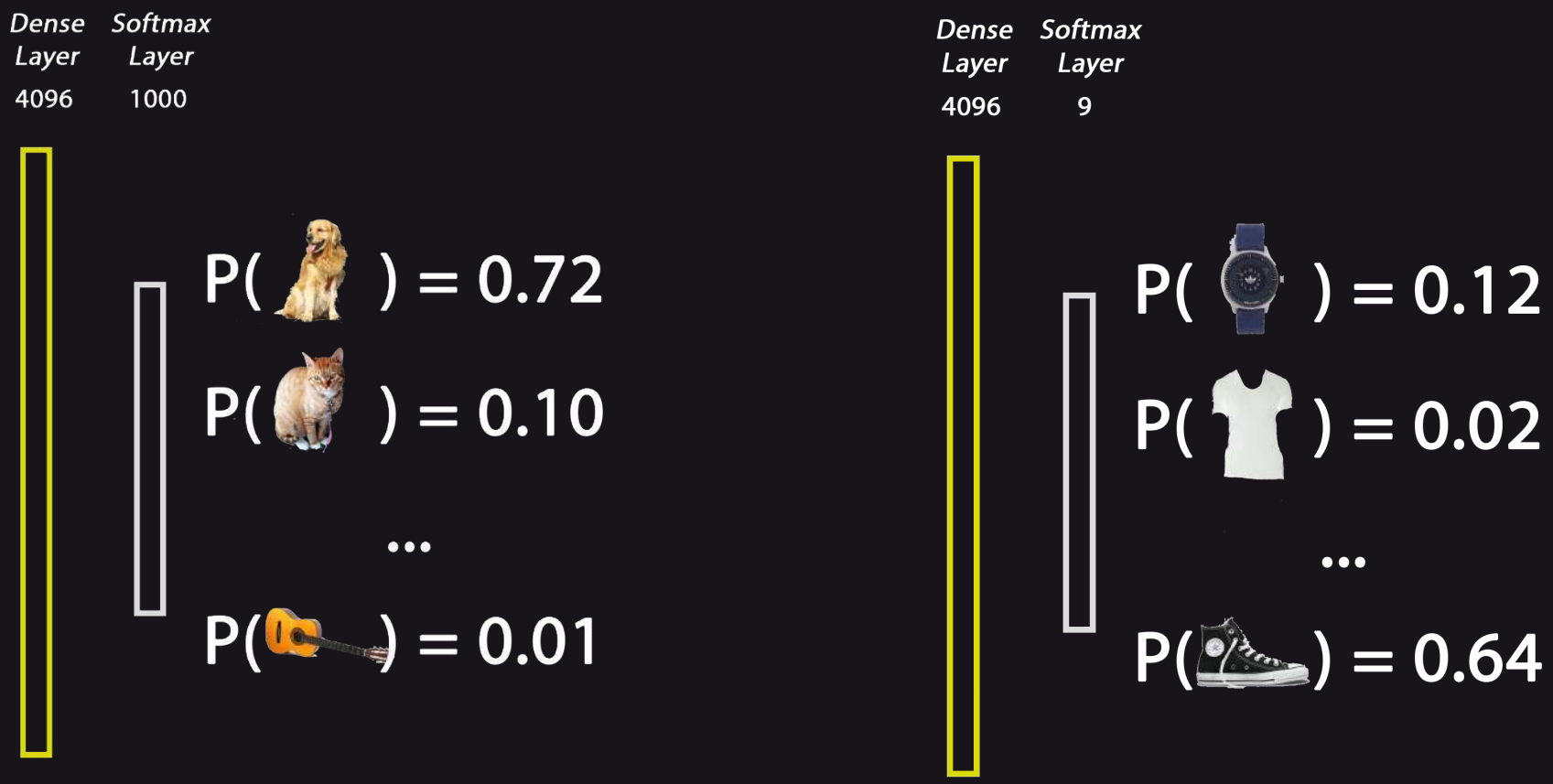
is a game changer

Fine-tuning A CNN To Solve A New Problem

- **Cut off last layer of pre-trained Imagenet winning CNN**
- **Keep learned network (convolutions) but replace final layer**
- **Can learn to predict new (completely different) classes**
- **Fine-tuning is re-training new final layer - learn for new task**



Fine-tuning A CNN To Solve A New Problem



Before Fine-Tuning

```
convolution2d_25 (Convolution2D) (None, 512, 14, 14)   2359808     zeropadding2d_25[0][0]
zeropadding2d_26 (ZeroPadding2D) (None, 512, 16, 16)   0           convolution2d_25[0][0]
convolution2d_26 (Convolution2D) (None, 512, 14, 14)   2359808     zeropadding2d_26[0][0]
maxpooling2d_10 (MaxPooling2D)   (None, 512, 7, 7)     0           convolution2d_26[0][0]
flatten_2 (Flatten)              (None, 25088)         0           maxpooling2d_10[0][0]
dense_4 (Dense)                  (None, 4096)          102764544   flatten_2[0][0]
dropout_3 (Dropout)              (None, 4096)          0           dense_4[0][0]
dense_5 (Dense)                  (None, 4096)          16781312    dropout_3[0][0]
dropout_4 (Dropout)              (None, 4096)          0           dense_5[0][0]
dense_6 (Dense)                  (Non , 1000)          4097000     dropout_4[0][0]
Total params: 138357544
```

After Fine-Tuning

```
# chop off last 2 layers
m.pop()
m.pop()
```

```
# now final layer is:
# dense_2 (Dense)
m.summary()
```

```
(Convolution2D) (None, 512, 14, 14)   0          zeropadding2d_38[0][0]
(ZeroPadding2D) (None, 512, 16, 16)   0          convolution2d_38[0][0]
(Convolution2D) (None, 512, 14, 14)   0          zeropadding2d_39[0][0]
maxpooling2d_15 (MaxPooling2D)   (None, 512, 7, 7)    0          convolution2d_39[0][0]
flatten_3 (Flatten)              (None, 25088)        0          maxpooling2d_15[0][0]
dense_9 (Dense)                  (None, 4096)         0          flatten_3[0][0]
dropout_5 (Dropout)              (None, 4096)         0          dense_9[0][0]
dense_10 (Dense)                 (None, 4096)         0          dropout_5[0][0]
dropout_6 (Dropout)              (None, 4096)         0          dense_10[0][0]
dense_12 (Dense)                 (None, 9)            36873      dropout_6[0][0]
```

Fine-tuning A CNN To Solve A New Problem

- **Fix weights in convolutional layers (set trainable=False)**
- **Remove final dense layer that predicts 1000 ImageNet classes**
- **Replace with new dense layer to predict 9 categories**

```
vgg.model.optimizer.lr = 0.001
model_run+=1
vgg.fit(batches, val_batches, nb_epoch=1)
vgg.model.save_weights(results_path + "finetuned_%d.h5" % model_run)
```

```
Epoch 1/1
3795/3795 [==============================] - 109s - loss: 0.6113 - acc: 0.8192 - val_loss: 0.3611
 - val_acc: 0.8804
```

88% accuracy in under 2 minutes for classifying products into categories

Fine-tuning is awesome!

Insert obligatory brain analogy

https://memeburn.com/2017/06/spree-image-search/

memeburn™ SUBSCRIBE ABOUT CONTACT ADVERTISE BURN MEDIA

WORLD | INDUSTRY NEWS | GEEK CULTURE | SOCIAL | FUTURE TECH | JOBS

Future Tech / Artificial Intelligence /

Spree adds AI-powered image search to iOS app

By Andy Walker: Editor on 8 June, 2017 *AndyWalkerSA*

Visual Similarity

https://memeburn.com/2017/06/spree-image-search/

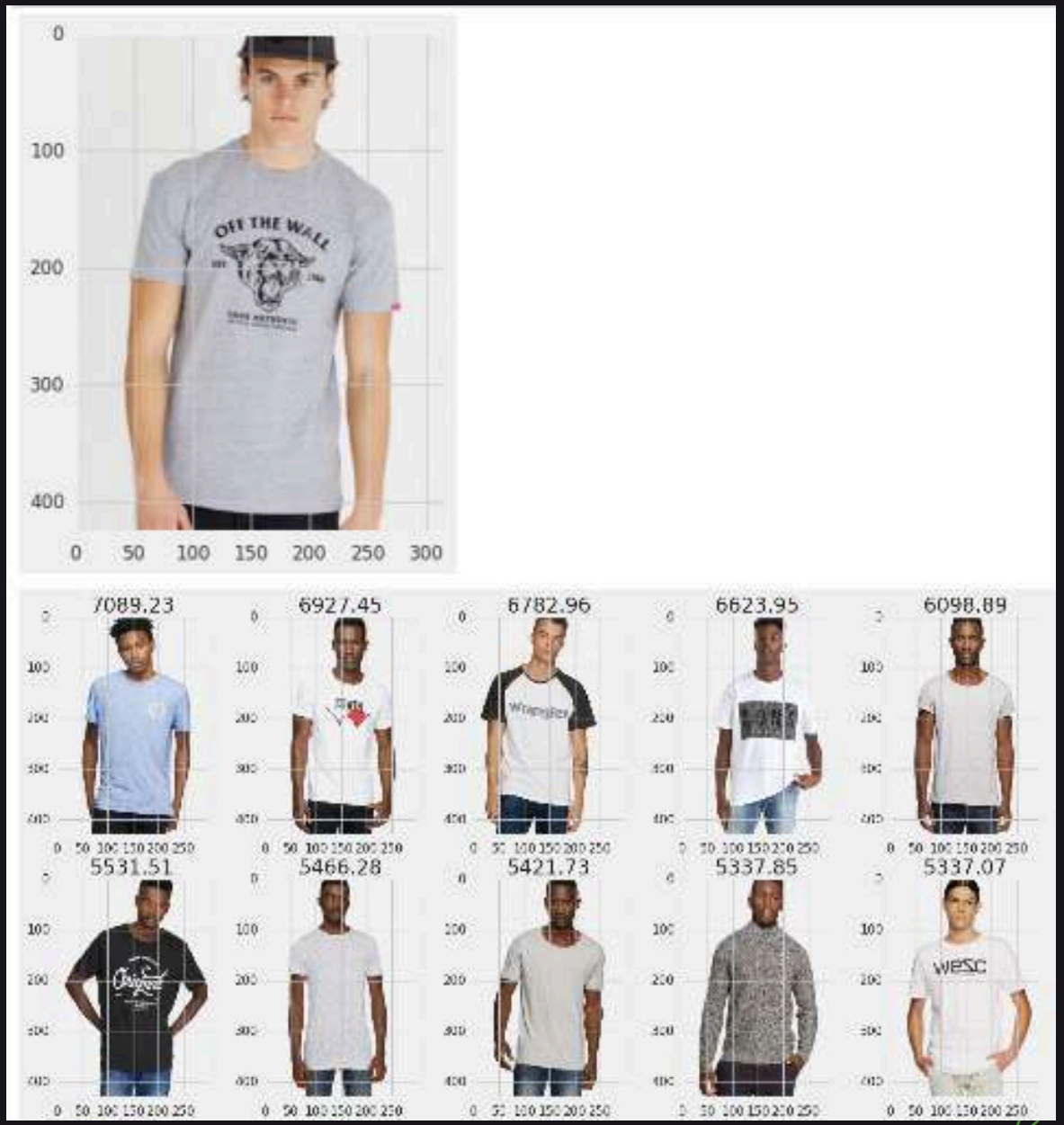
- **Chop off last 2 VGG layers**
- **Use dense layer with 4096 activations**
- **Compute nearest neighbours in the space of these activations**

https://github.com/alexcnwy/CTDL_CNN_TALK_20170620

```
def get_most_similar_products(test_img_idx):
    # plot test img
    vec_test.loc[test_img_idx]['filename']
    plotimg(pwd + '/data/test/' + vec_test.loc[test_img_idx]['filename'])

    # do dot prod
    test_img_vec = vec_test_num.loc[test_img_idx][:]
    test_img_vec = test_img_vec.reshape(len(test_img_vec),1)
    a = np.dot(vec_valid_num.ix[:], test_img_vec)

    # transform scores
    results = pd.DataFrame(a)
    results['filenames'] = vec_valid['filename']
    results.columns = ['scores', 'filenames']
    results.sort_values('scores', ascending = False, inplace = True)
    results.head()

    # get matches
    matches = results['filenames'].values[:10]
    match_scores = results['scores'].values[:10]

    plot_pic_grid(matches)
```

Input Image

not seen by model

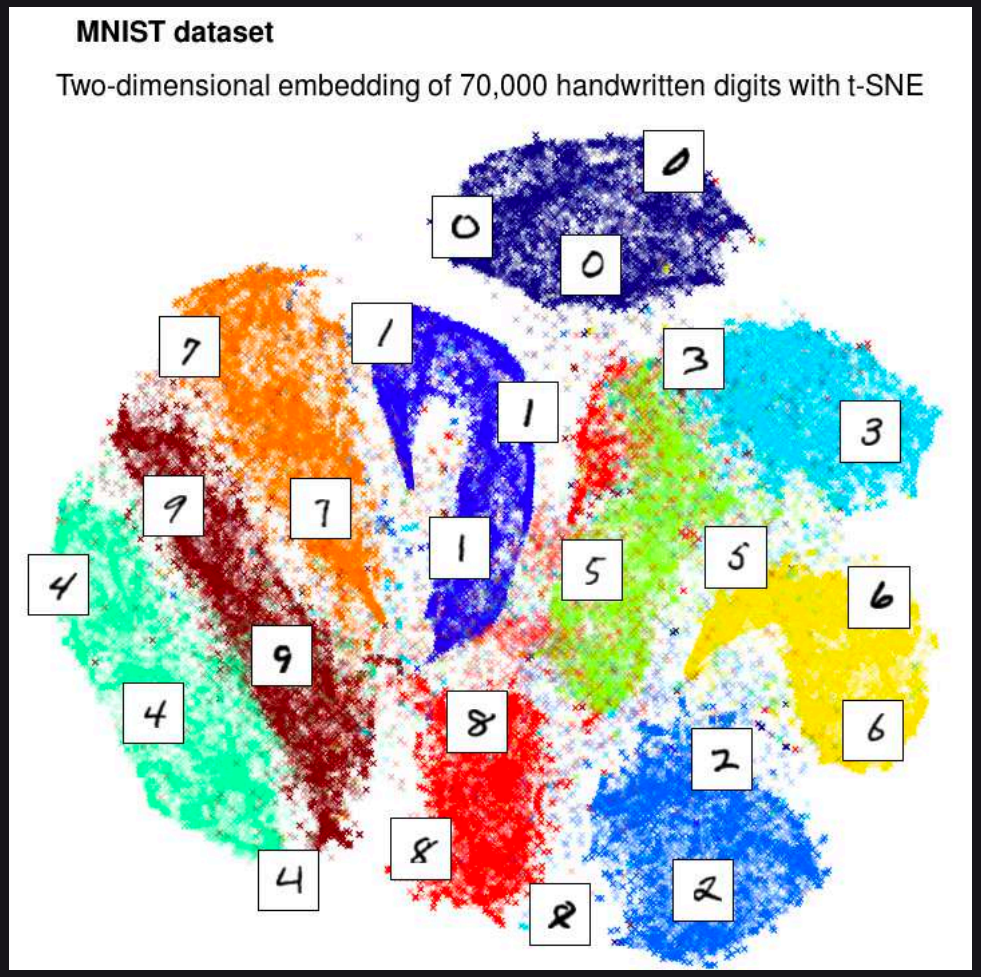


Results

Top 10 most "visually similar"

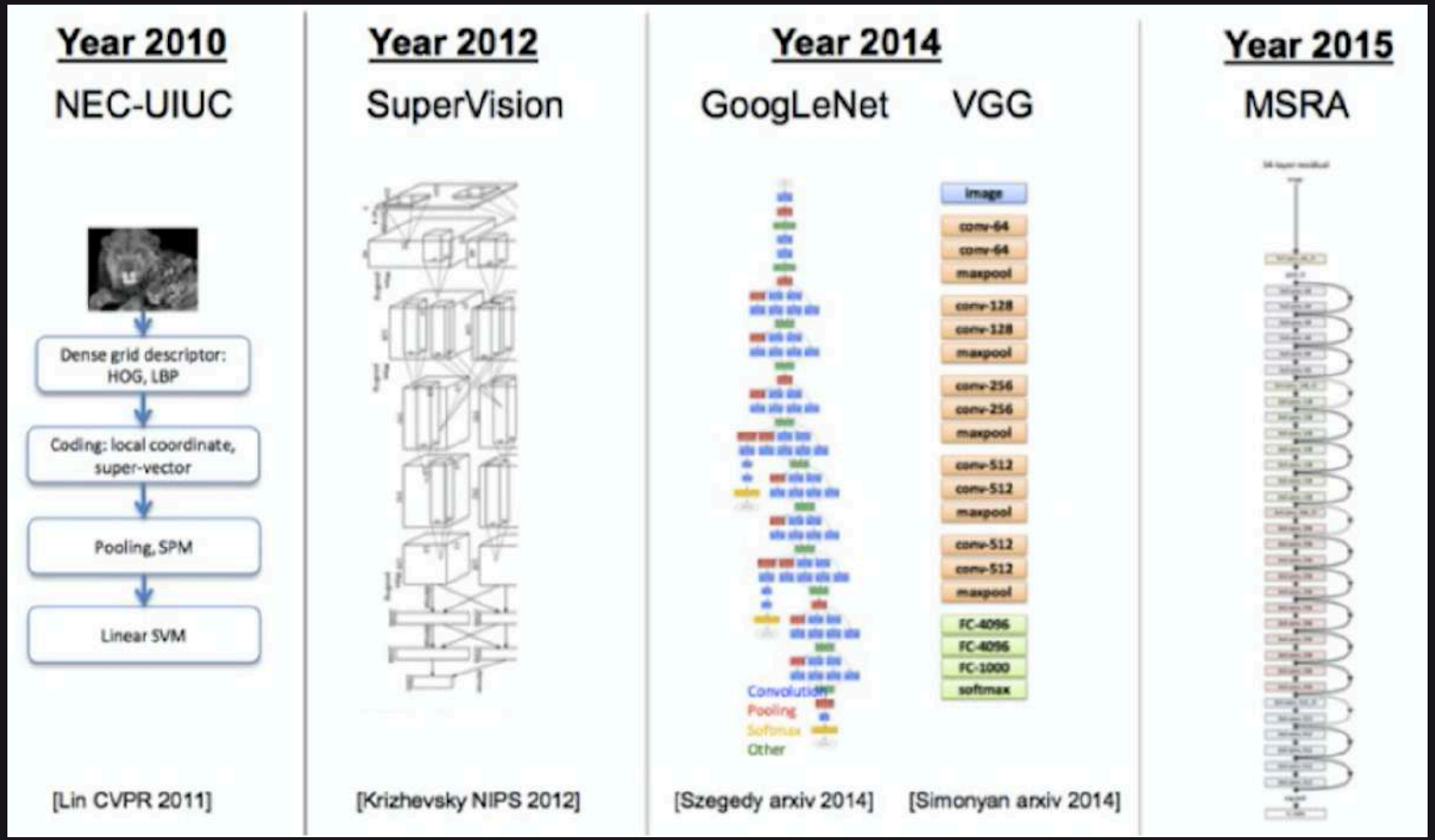
Final Convolutional Layer = Semantic Vector

- **The final convolutional layer encodes everything the network needs to make predictions**
- **The dense layer added on top and the softmax layer both have lower dimensionality**



4. More Advanced Methods

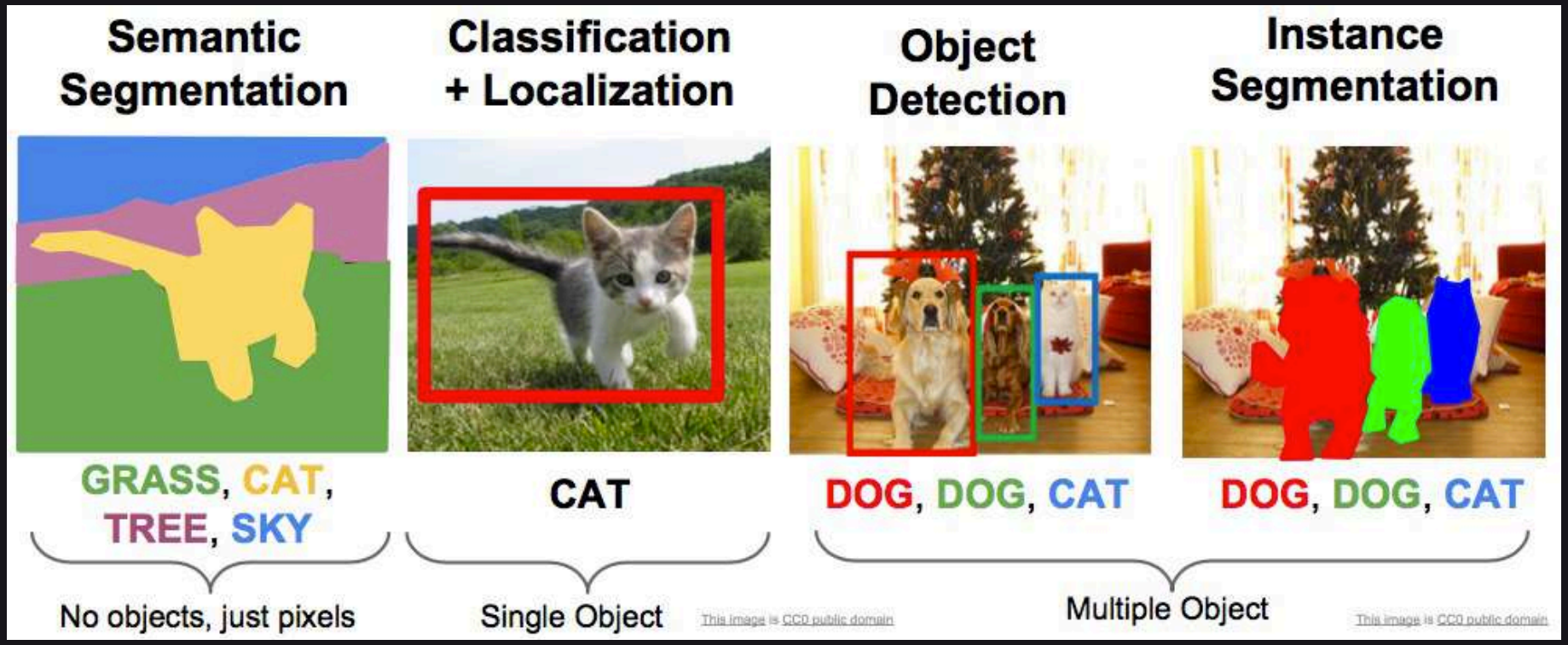
Use a Better Architecture (or all of them!)



"Ensembles win"

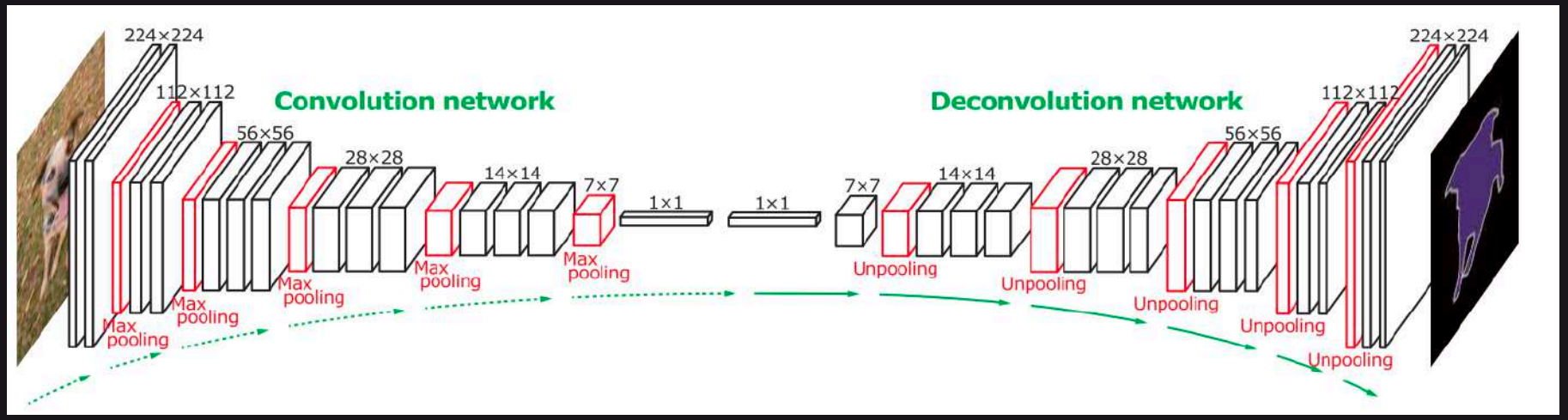
learn a weighted average of many models' predictions

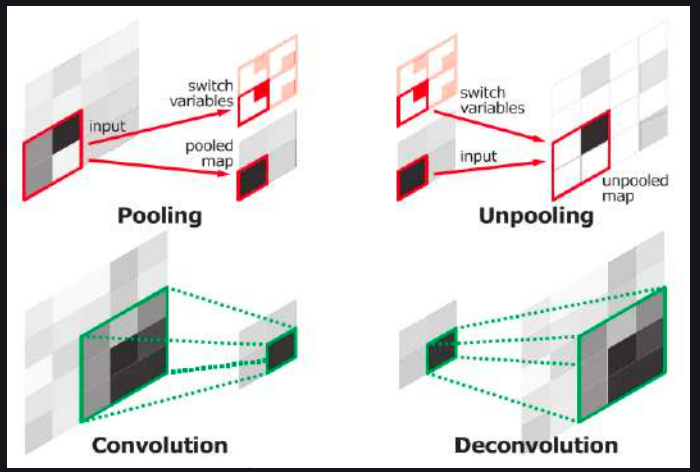
There are MANY Computer Vision Tasks



cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

Semantic Segmentation

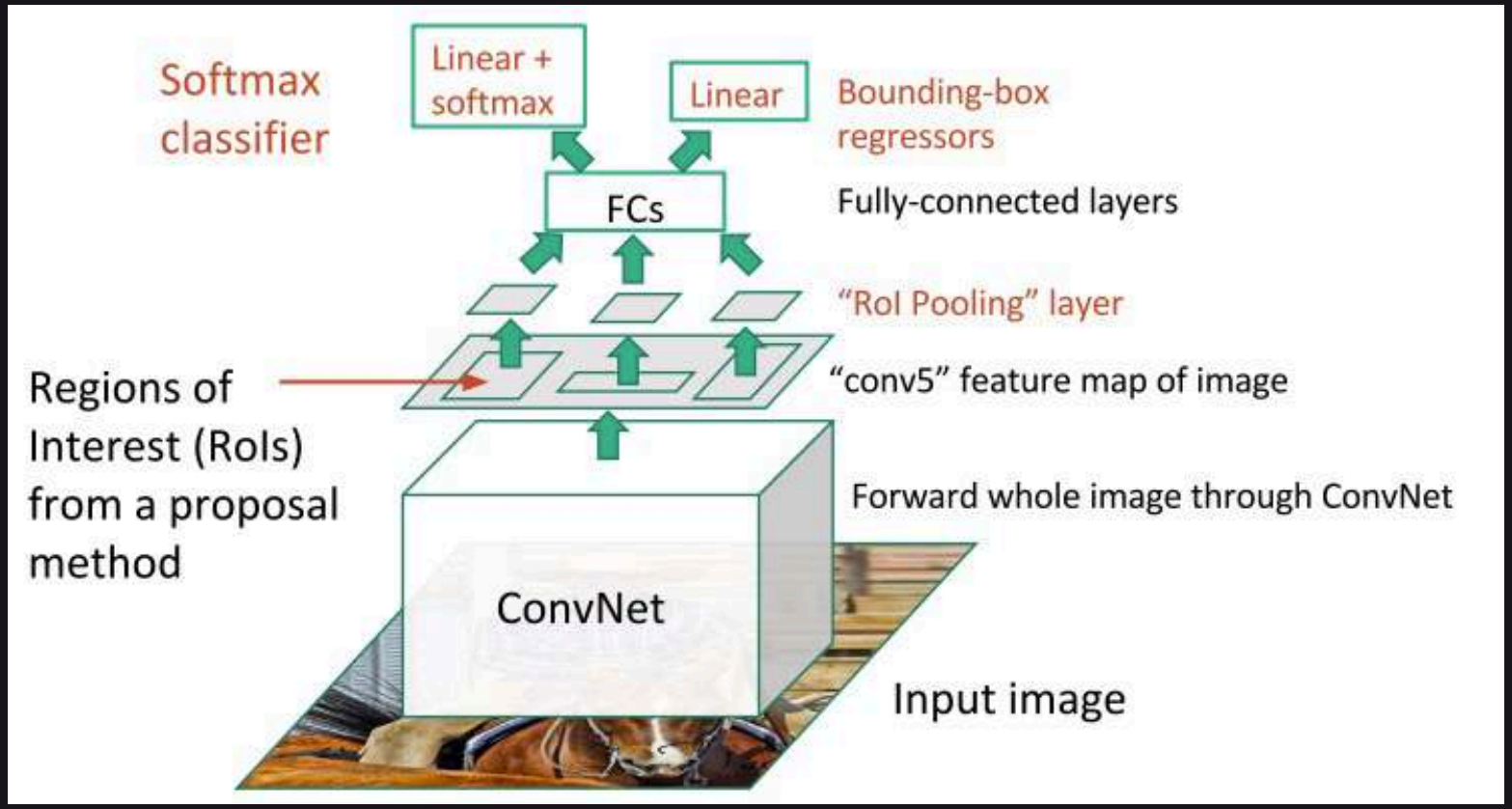




Long et al. “Fully Convolutional Networks for Semantic Segmentation” CVPR 2015

Noh et al. Learning Deconvolution Network for Semantic Segmentation. IEEE on Computer Vision 2016

Object detection



http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

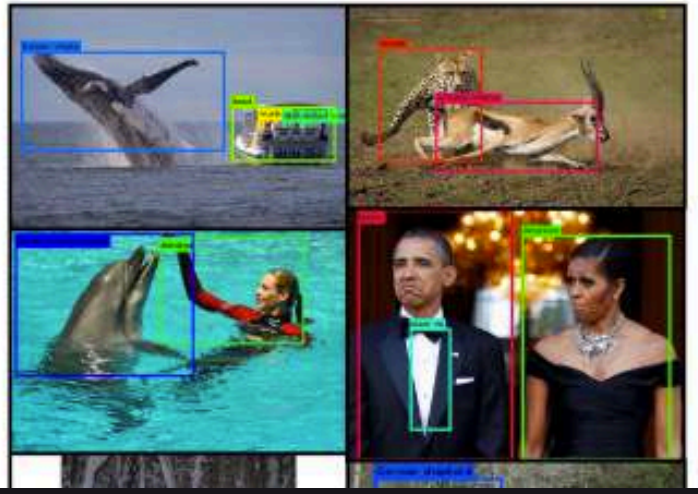
Object detection

YOLO9000:
Better, Faster, Stronger

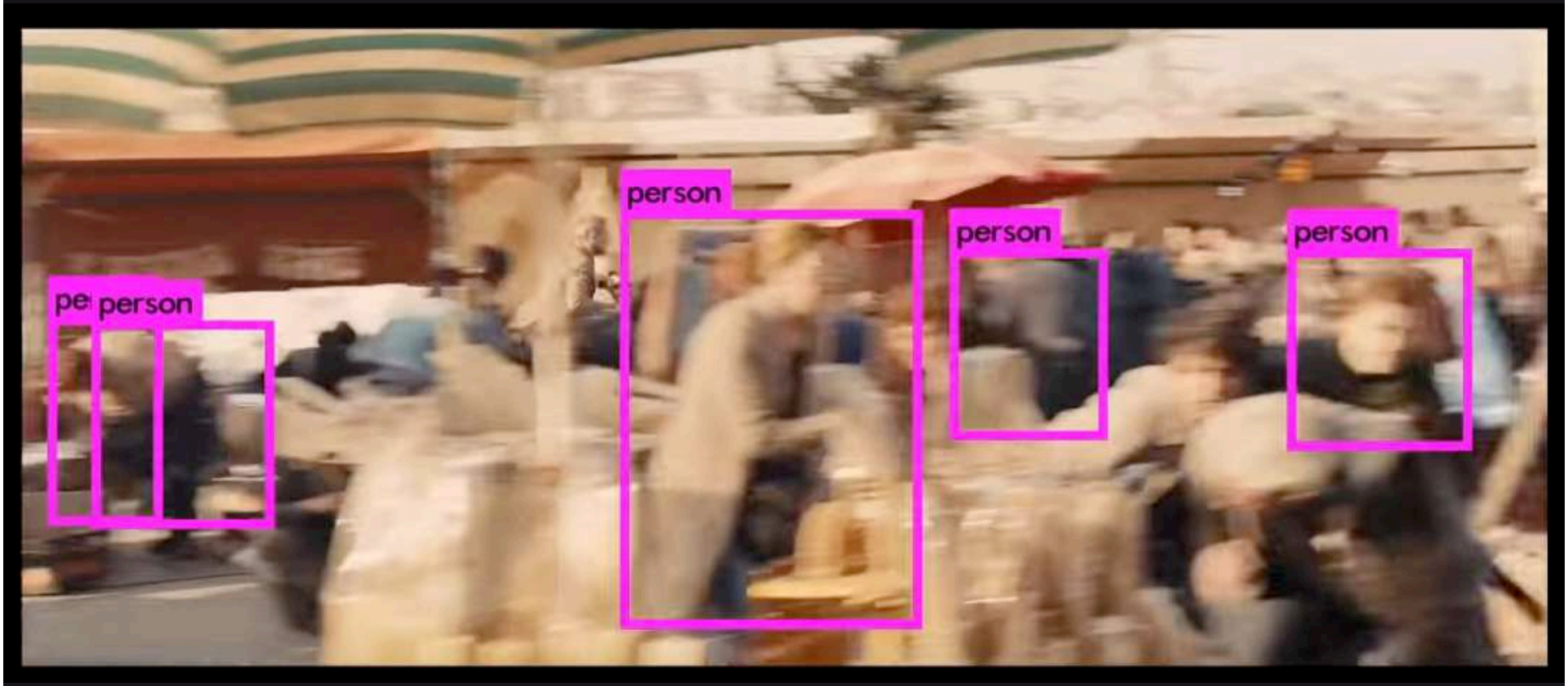
Joseph Redmon*†, Ali Farhadi*†
University of Washington*, Allen Institute for AI†
http://pjreddie.com/yolo9000/

Abstract

We introduce YOLO9000, a state-of-the-art, real-time object detection system that can detect over 9000 object categories. First we propose various improvements to the YOLO detection method, both novel and drawn from prior work. The improved model, YOLOv2, is state-of-the-art on standard detection tasks like PASCAL *VOC and COCO. Using a novel, multi-scale training method the same YOLOv2 model can run at varying sizes, offering an easy tradeoff between speed and accuracy. At 67 FPS, YOLOv2 gets 76.8 mAP on VOC 2007. At 40 FPS, YOLOv2 gets 78.6 mAP, outperforming state-of-the-art methods like Faster R-CNN with ResNet and SSD while still running significantly faster. Finally we propose a method to jointly train on ob-*



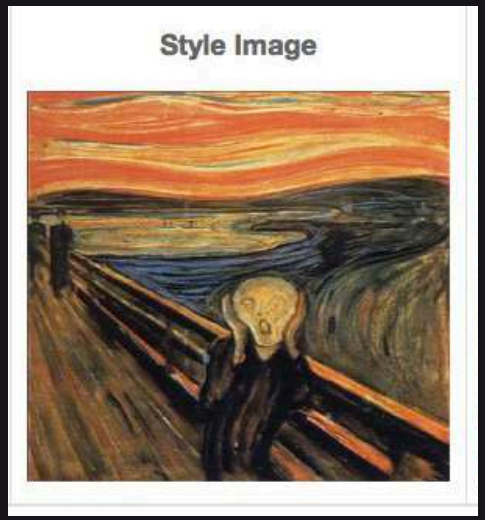
Object detection



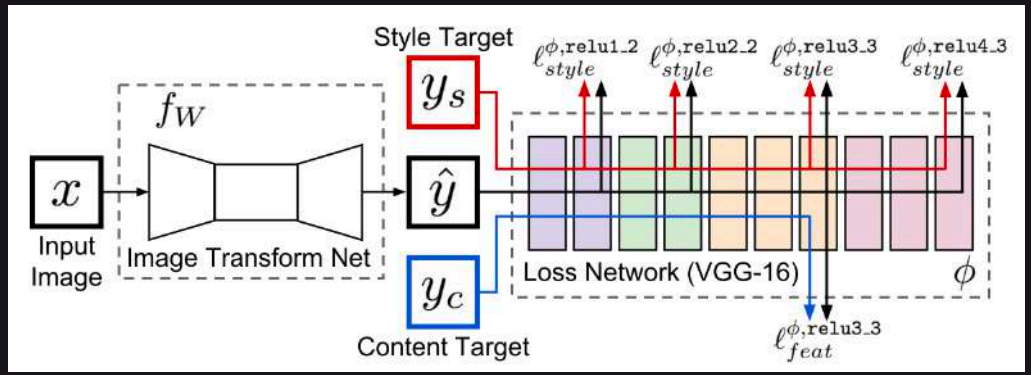
https://www.youtube.com/watch?v=VOC3huqHrss

Style Transfer

f (

, 

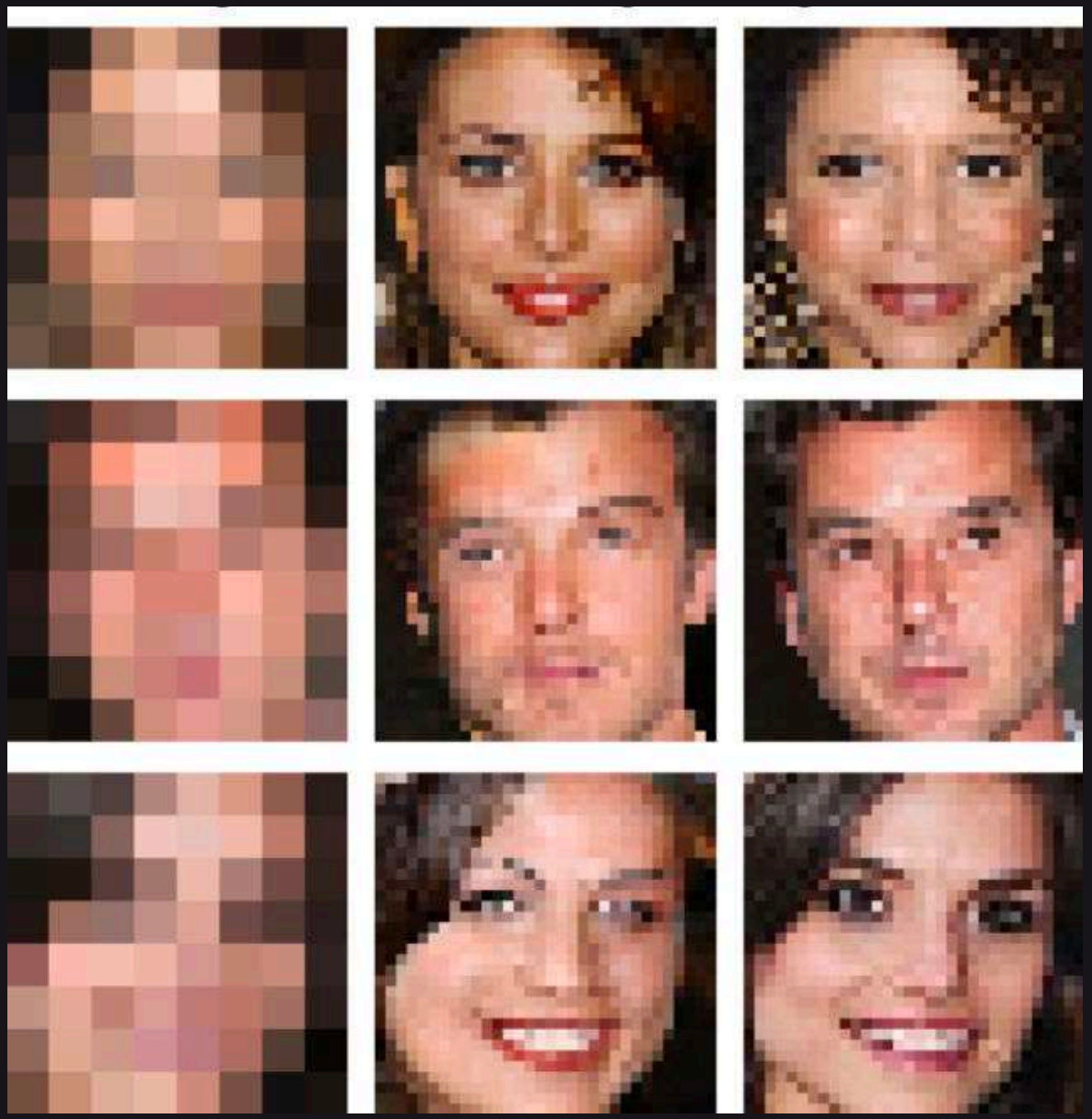
) = 



http://blog.romanofoti.com/style_transfer/
Johnson et al. Perceptual losses for real-time style transfer and super-resolution. 2016

https://www.youtube.com/watch?v=LhF_56SxrGk

Pixelated Output Original

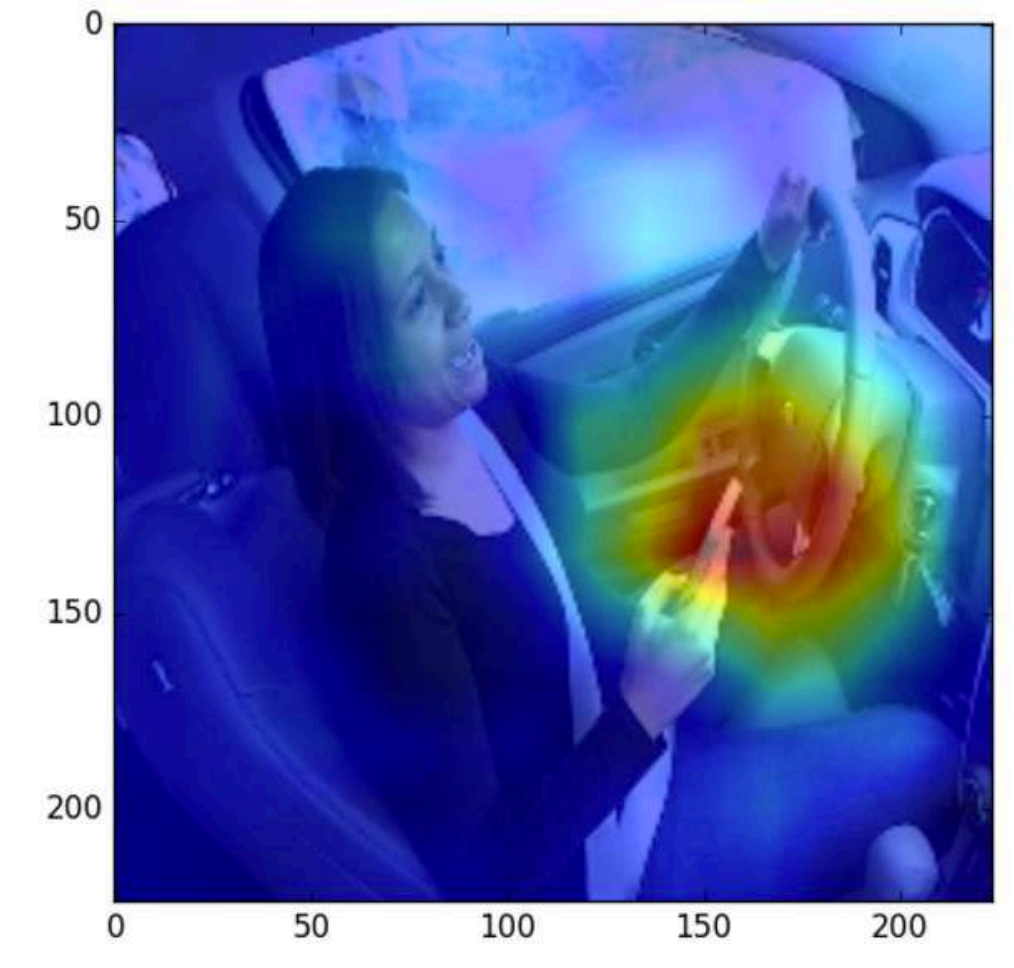


https://arstechnica.com/information-technology/2017/02/google-brain-super-resolution-zoom-

Super-Resolution

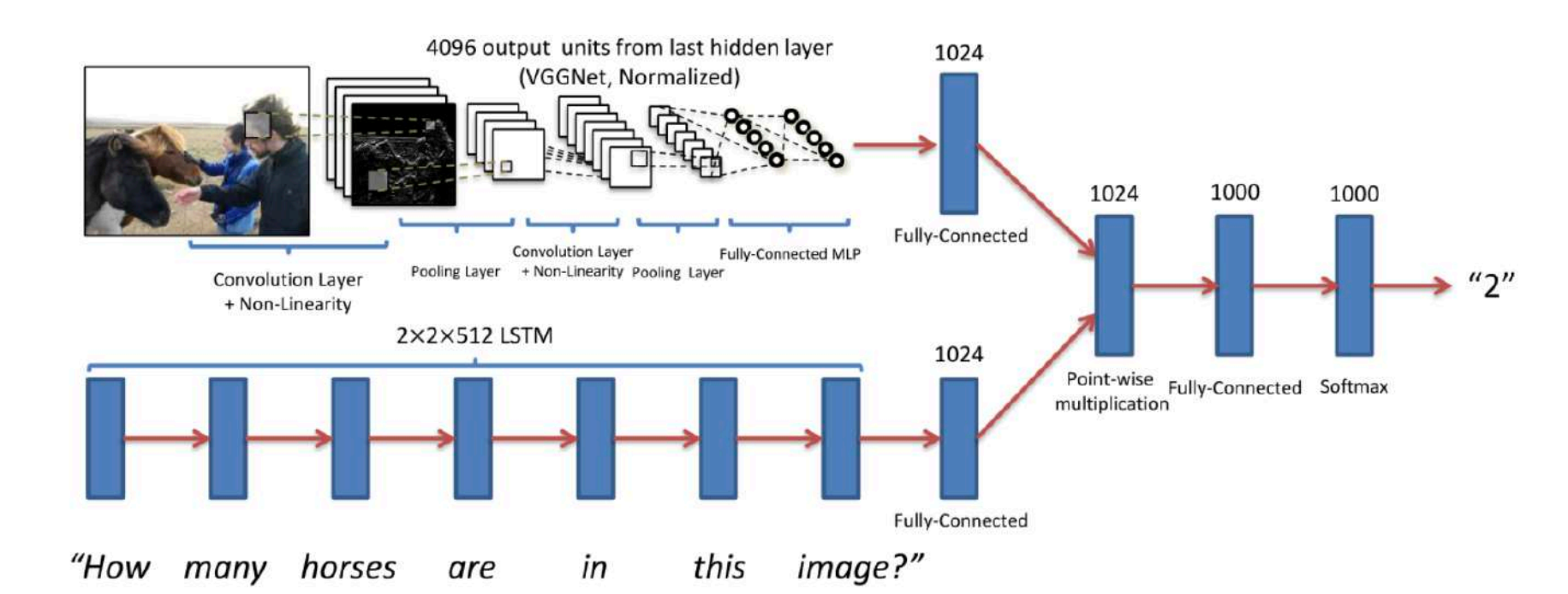
This image is
3.8 kb

Visual Attention



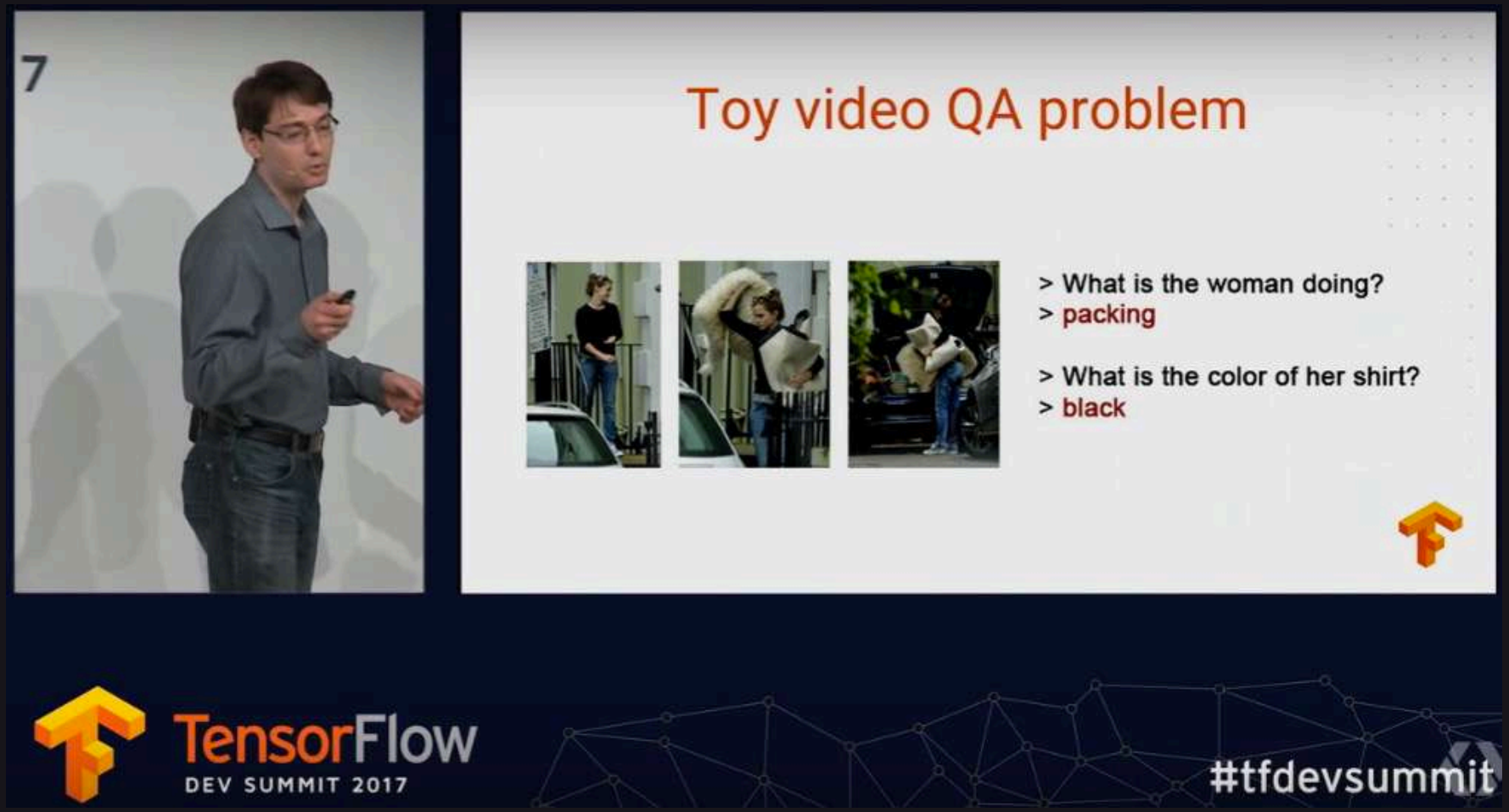
https://github.com/tdeboissiere/BGG16CAM-keras

Image Q&A



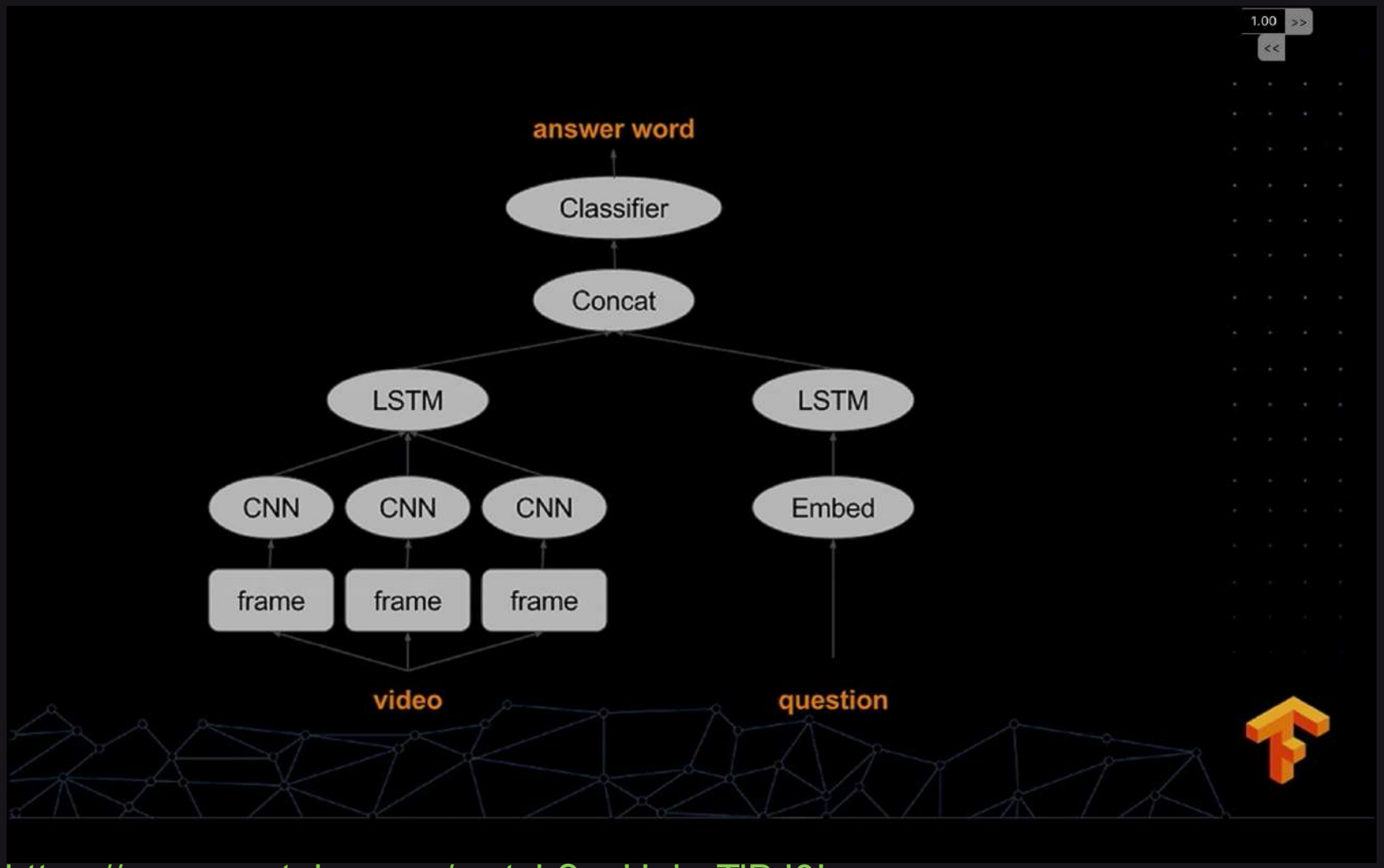
http://iamaaditya.github.io/2016/04/visual_question_answering_demo_notebook

Video Q&A



https://www.youtube.com/watch?v=UeheTiBJ0Io

Video Q&A



https://www.youtube.com/watch?v=UeheTiBJ0Io

5. Case Studies

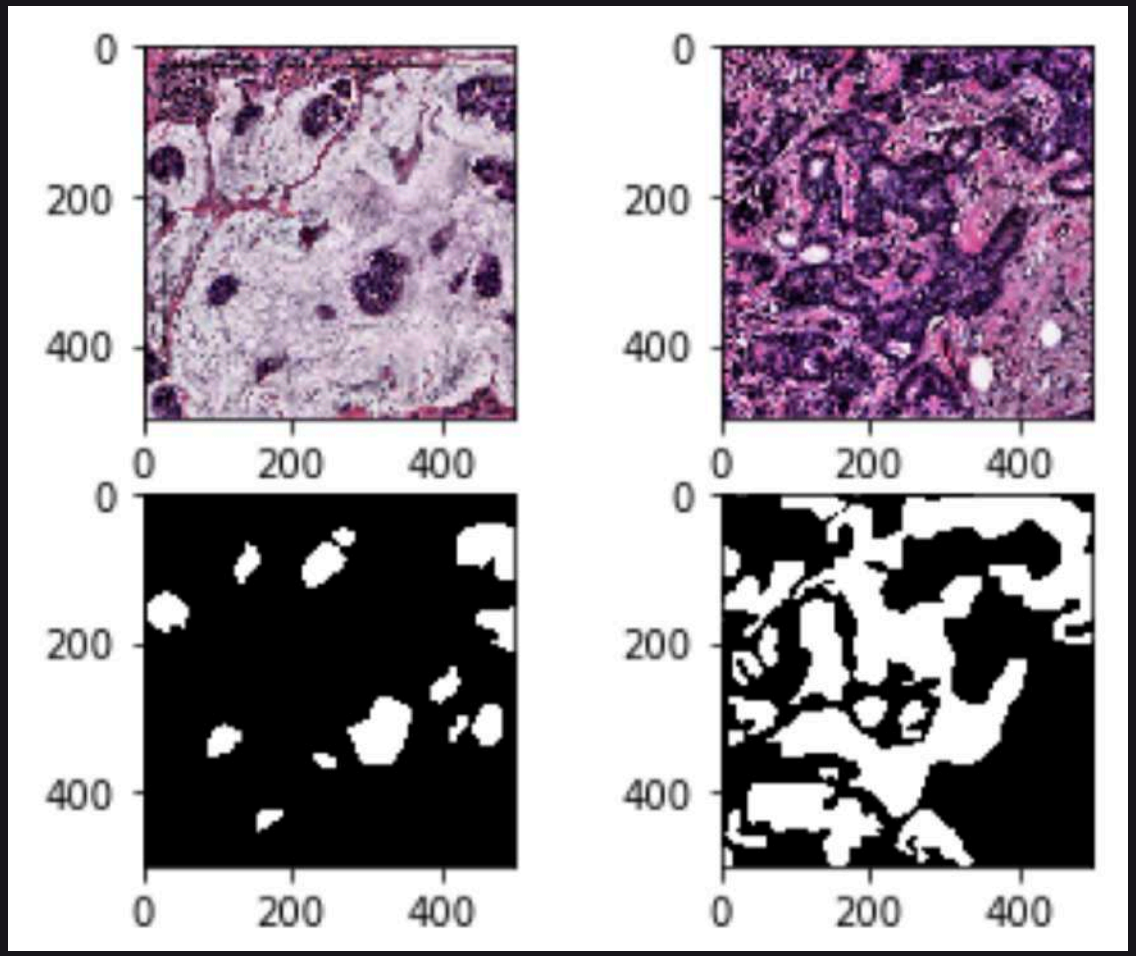
Estimating Accident Repair Cost from Photos

Prototype for large SA insurer

Detect car make & model from registration disk

Predict repair cost using learned

Segmenting Medical Images



Counting People

Count shoppers, segment on age & gender

facial recognition loyalty is next

Detecting Potholes

Extracting Data from Product Catalogues

Real-time ATM Video Classification

Optical Sorting

https://www.youtube.com/watch?v=Xf7jaxwnyso

https://www.autoblog.com/2017/08/04/self-driving-car-sign-hack-stickers/