Deep Neural Networks for Video Applications at the Edge

Alex Conway
@ALXCNWY
alex @ numberboost.com
NUMBERBOOST

Oslo Machine Learning Meetup 2019.05.16

NUMBERBOOST

we build custom A.I. solutions

2016 MultiChoice
Innovation Competition
1st Prize Winners

2017 Mercedes-Benz
Innovation Competition
1st Prize Winners

2018 Lloyd's Register
Innovation Competition
1st Prize Winners

2019 NTT & Dimension Data
Innovation Competition
1st Prize Winners

Meetup

Cape Town Machine Learning Meetup

Cape Town, South Africa · 660 members · Public group

Cape Town Deep Learning

Cape Town, South Africa · 378 members · Public group

Machine Intelligence Institute of Africa
Transform Africa through Machine Intelligence

DSN
Data Science Nigeria


JUST HODL IT

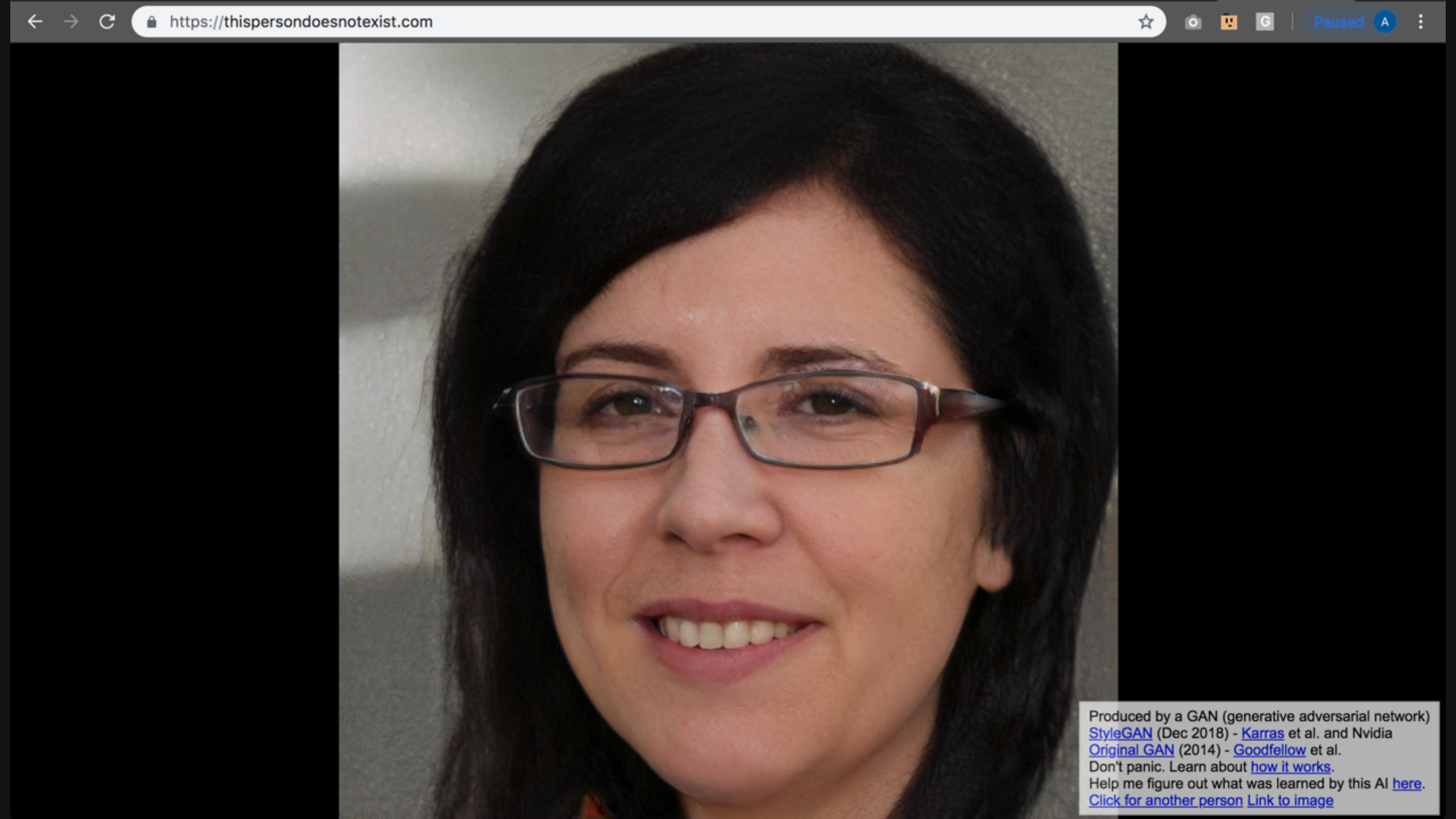
HANDS UP!

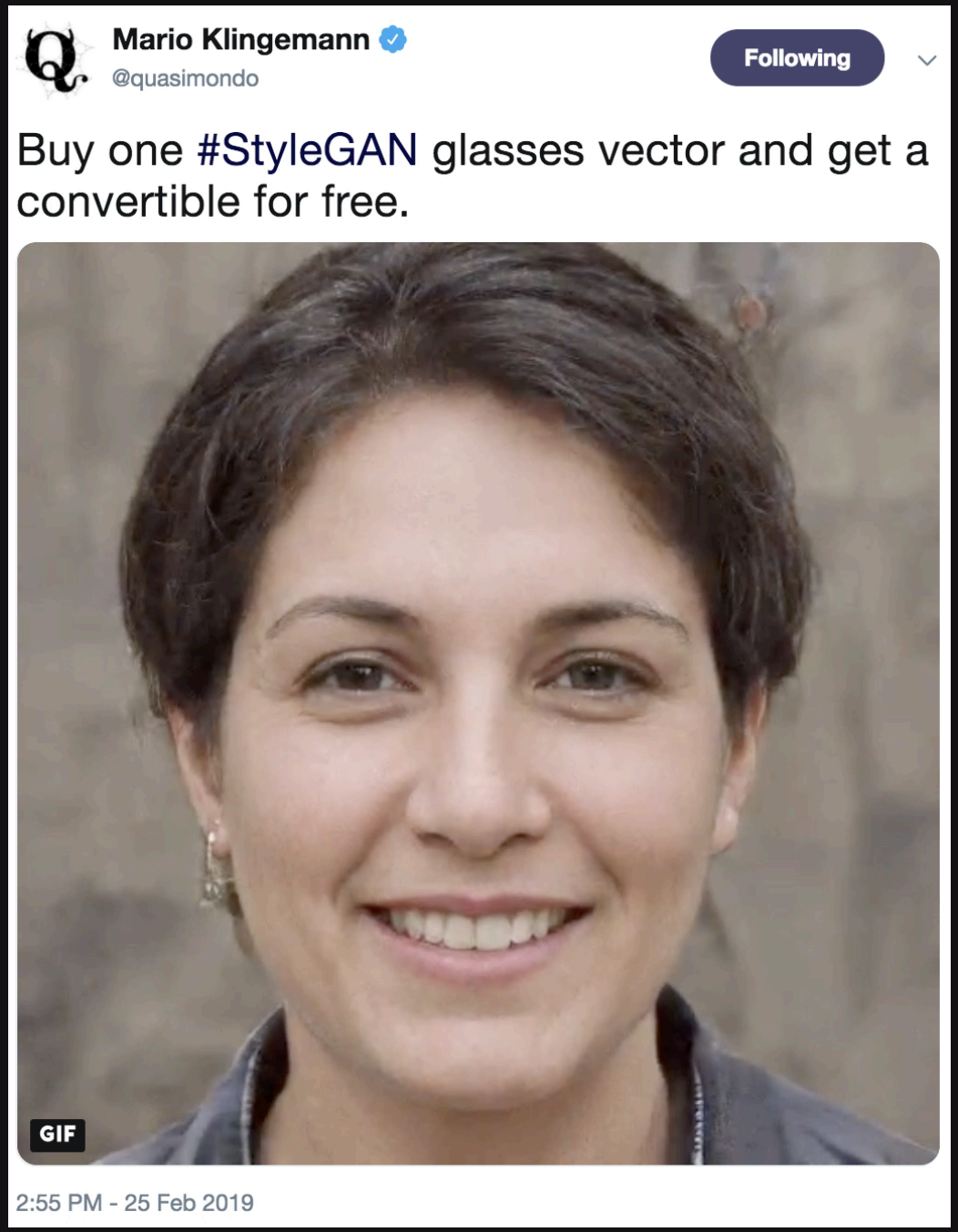

MAGIC

Image Credit: NVIDIA

https://twitter.com/goodfellow_ian/status/1084973596236144640

WWW.THISPERSONDOESNOTEXIST.COM


https://thispersondoesnotexist.com
Paused
Produced by a GAN (generative adversarial network)
StyleGAN (Dec 2018) - Karras et al. and Nvidia
Original GAN (2014) - Goodfellow et al.
Don't panic. Learn about how it works.
Help me figure out what was learned by this AI here.
Click for another person Link to image



https://twitter.com/quasimondo/status/1100016467213516801

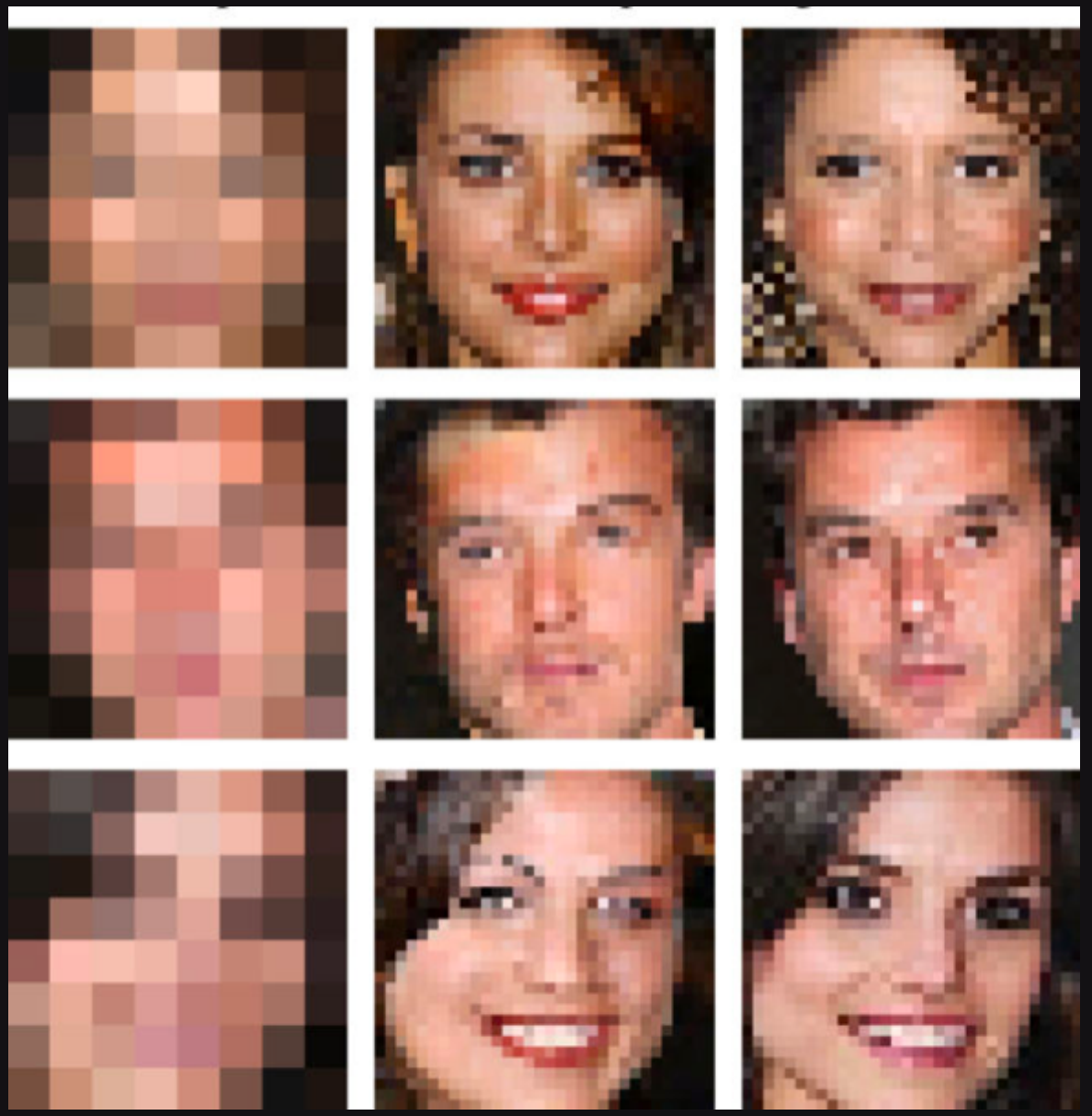
Twitter pays up to $150M for Magic Pony Technology, which uses neural networks to improve images

Ingrid Lunden @ingridlunden / 3 years ago

Comment

https://techcrunch.com/2016/06/20/twitter-is-buying-magic-pony-technology-which-uses-neural-networks-to-improve-images/

Pixelated Output Original



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

SUPER-RESOLUTION

This image is
3.8 kb

Original input

Rear Window (1954)

Pix2pix output

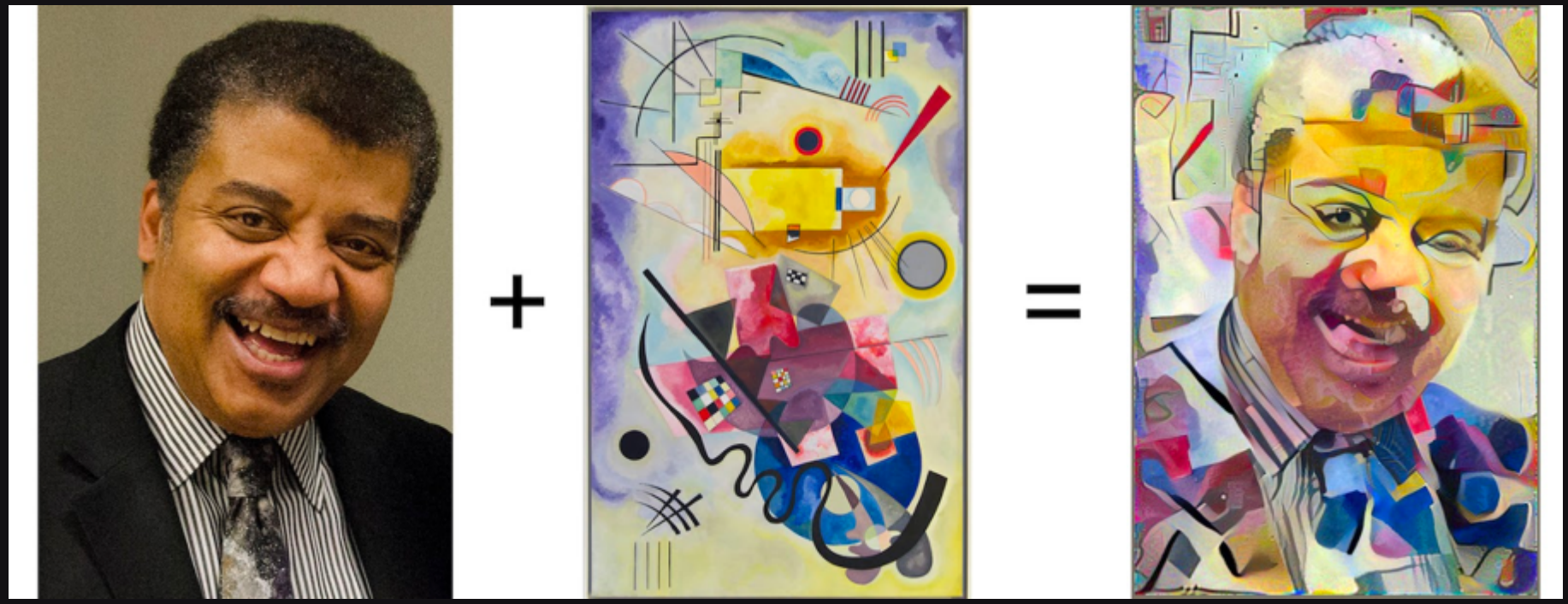
Fully Automated

Remastered

Painstakingly by Hand

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

STYLE TRANSFER



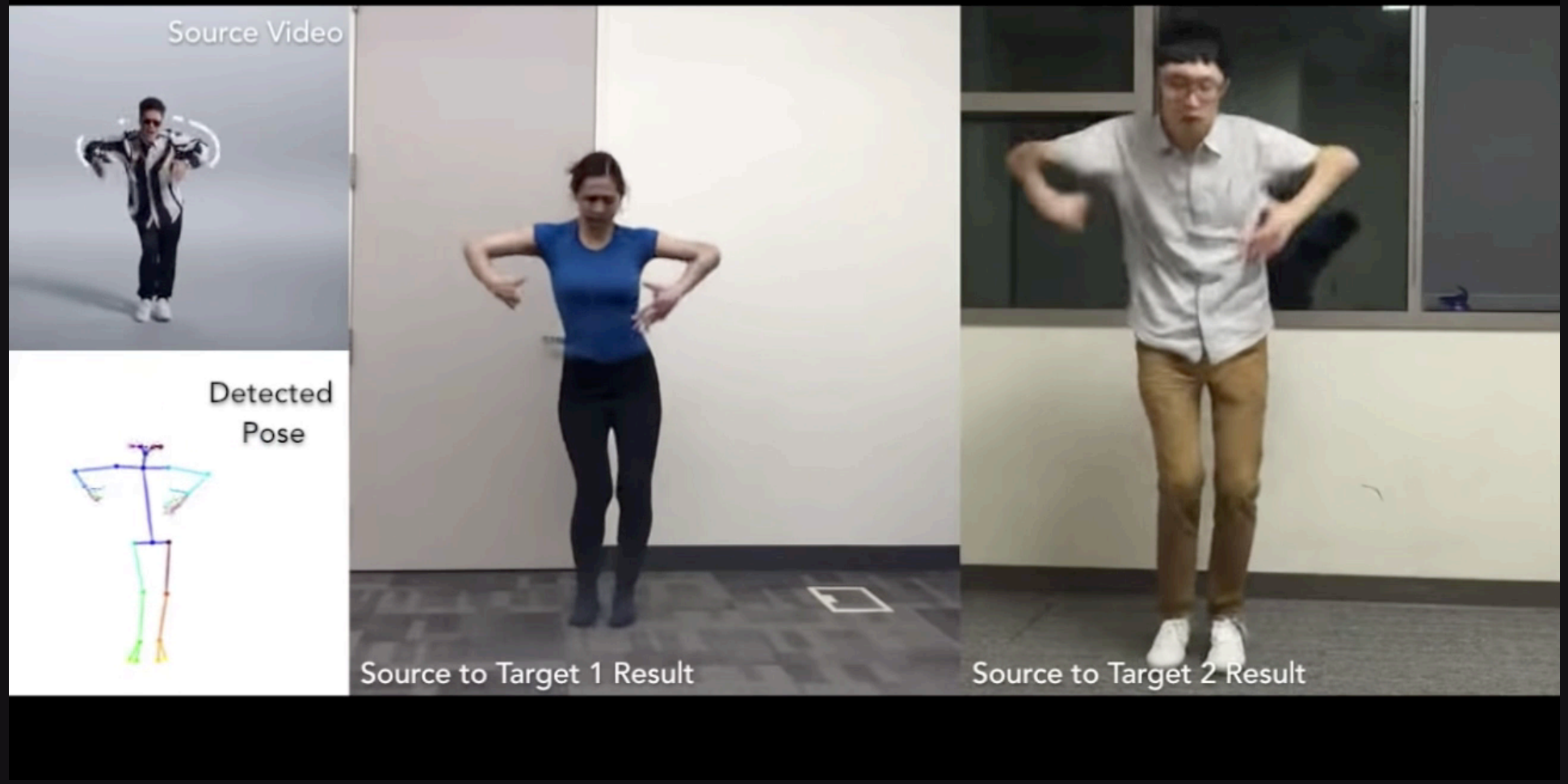
https://arxiv.org/abs/1508.06576

STYLE TRANSFER SORCERY

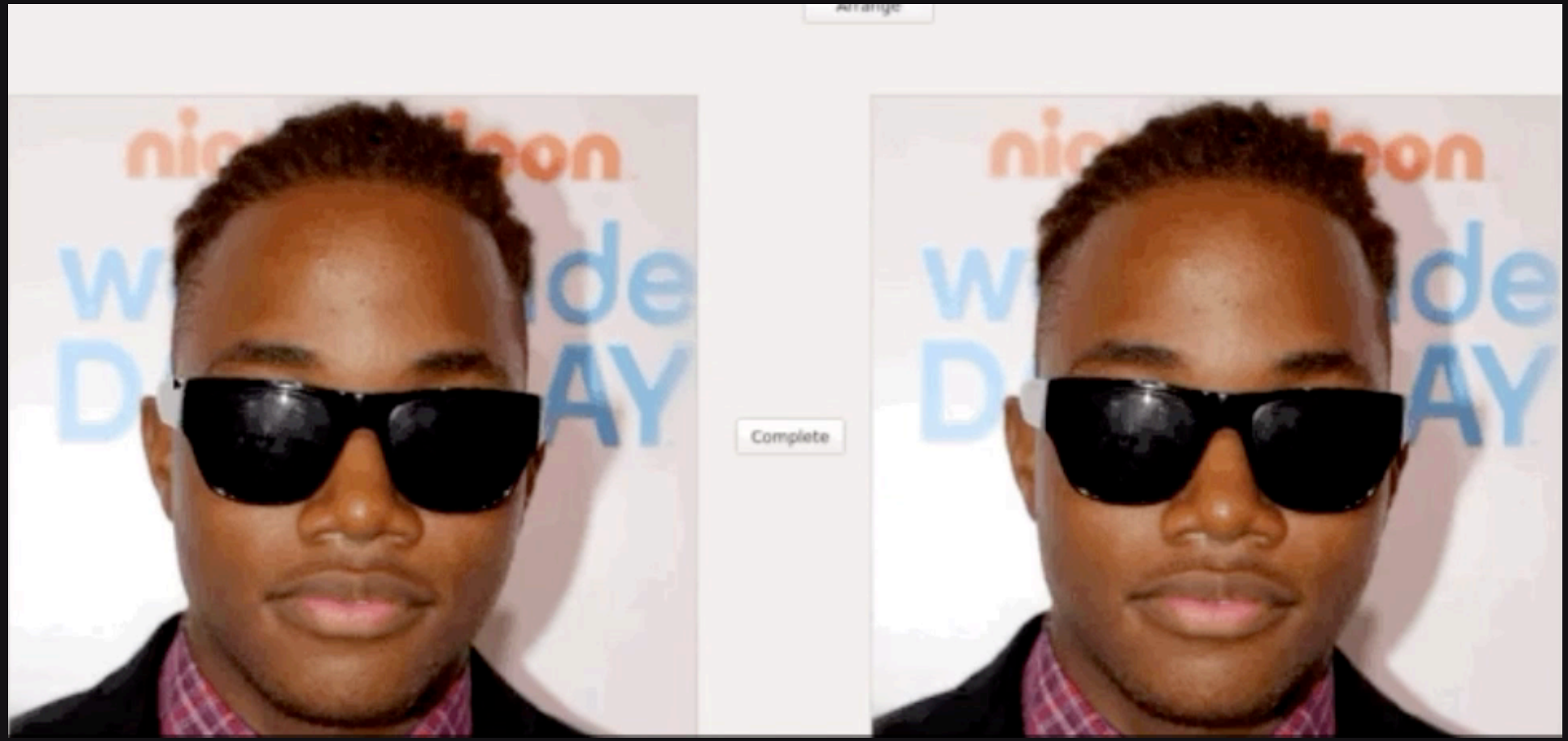
https://github.com/junyanz/CycleGAN

https://w.....top

https://www.linkedin.com/feed/update/urn:li:activity:6498172448196820993



https://news.developer.nvidia.com/ai-can-transform-anyone-into-a-professional-dancer/



https://github.com/JoYoungjoo/SC-FEGAN

MOTHERBOARD

DEEPFAKES | By Samantha Cole | Dec 11 2017, 9:18pm

AI-Assisted Fake Porn Is Here and We're All Fucked

Someone used an algorithm to paste the face of 'Wonder Woman' star Gal Gadot onto a porn video, and the implications are terrifying.

SHARE TWEET

https://motherboard.vice.com/en_us/article/gydydm/gal-gadot-fake-ai-porn

Family fun with deepfak
got my wife onto the To

Sven Charleer Follow

Feb 2, 2018 · 5 min read

https://towardsdatascience.com/family-fun-with-deepfakes-or-how-i-got-my-wife-onto-

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


成人
长裤
成人
短袖
非机动车
成人
短袖
短袖
长裤
成人
非机动车
男
成人
短袖
长裤
非机动车
成人
短袖
长裤

https://qz.com/1202075/chinese-police-are-using-f glasses-for-surveliance/

REAL FAKE NEWS

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

China Xinhua News

@XHNews

Follow

Xinhua's male #AI anchor can now broadcast news in a standing position, with more body language

VIDEO APPLICATIONS

- GANs / Video Generation
- **Clip** Classification
- **Frame** Classification
- **Object** Detection etc.
- Caption Generation
- Video Q & A

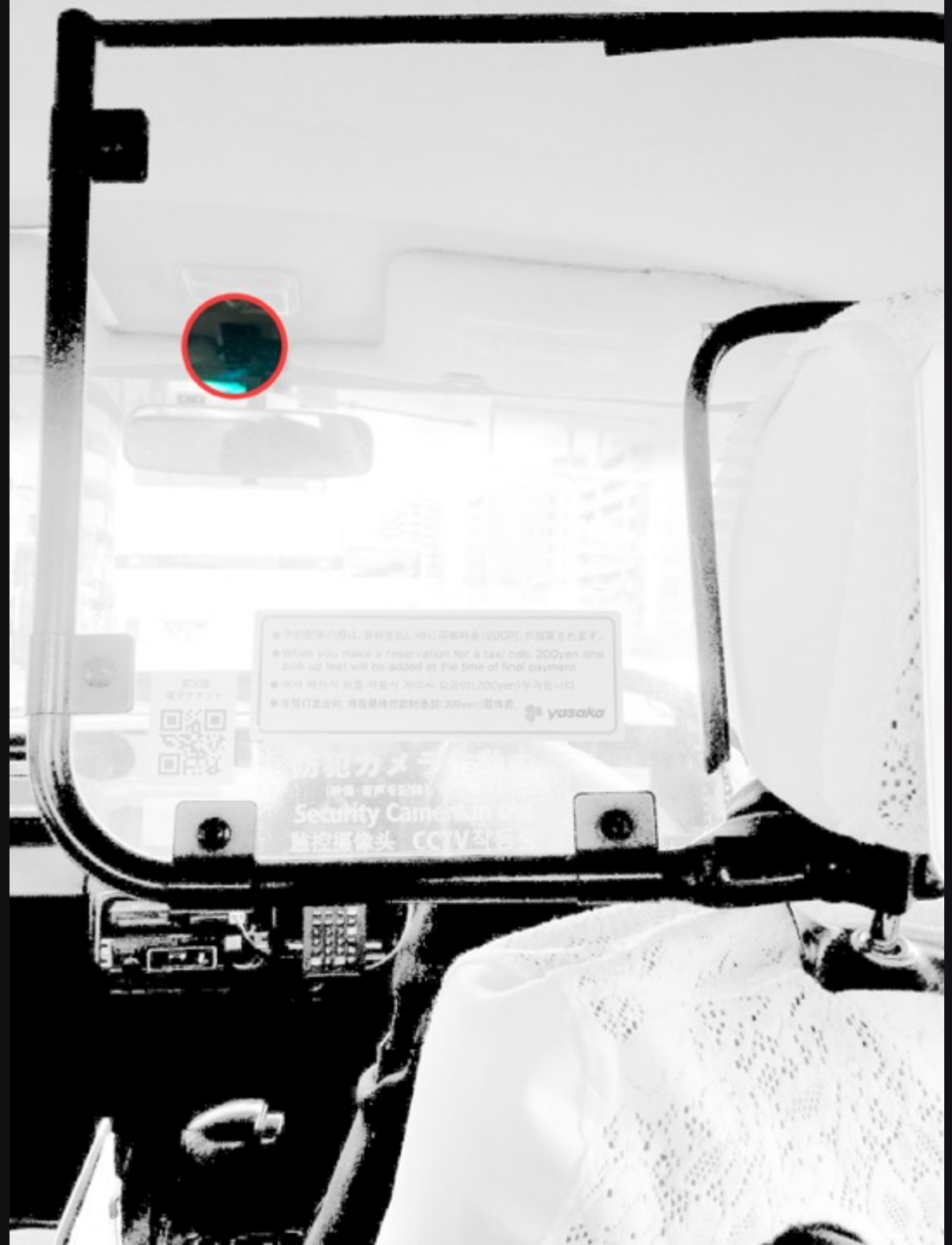
SIZE DOES MATTER

(w,h,3,t)

e.g.
500x500x3x4x60
= 180 million


JR線のりかえ口(南のりかえ口)
Transfer to JR Lines (South Transfer)


● 予約配車の際は、最終支払い時に迎車料金(200円)が加算されます。
● When you make a reservation for a taxi cab, 200yen (the pick up fee) will be added at the time of final payment.
● 예약 배차시 최종 지불시 게이사 요금이(200yen)부과됩니다.
● 在预订发出时，将在最终付款时添加(200yen)取件费。
yasaka
Security Camera in use
监控摄像头 CCTV작동중


yasaka


Polo
PABLO
PABLO
PABLO


Polo


防
カメラ作動中
消火器
上り
エスカレーターは
段対側へお回り
ください。
新い金麦
金麦

1. NEURAL NETWORK CRASH COURSE

2. CONVOLUTIONAL NEURAL NETWORKS

3. RECURRENT NEURAL NETWORKS

4. OBJECT DETECTION

5. SHOW AND TELL

6. EDGE COMPUTING & FEDERATED ML

7. SEBEN2.AI

GREAT FREE RESOURCES

Jeremy Howard & Rachel Thomas

http://course.fast.ai

Richard Socher's Class on NLP (great RNN resource)

http://web.stanford.edu/class/cs224n/

Andrej Karpathy's Class on Computer Vision (great CNN resource)

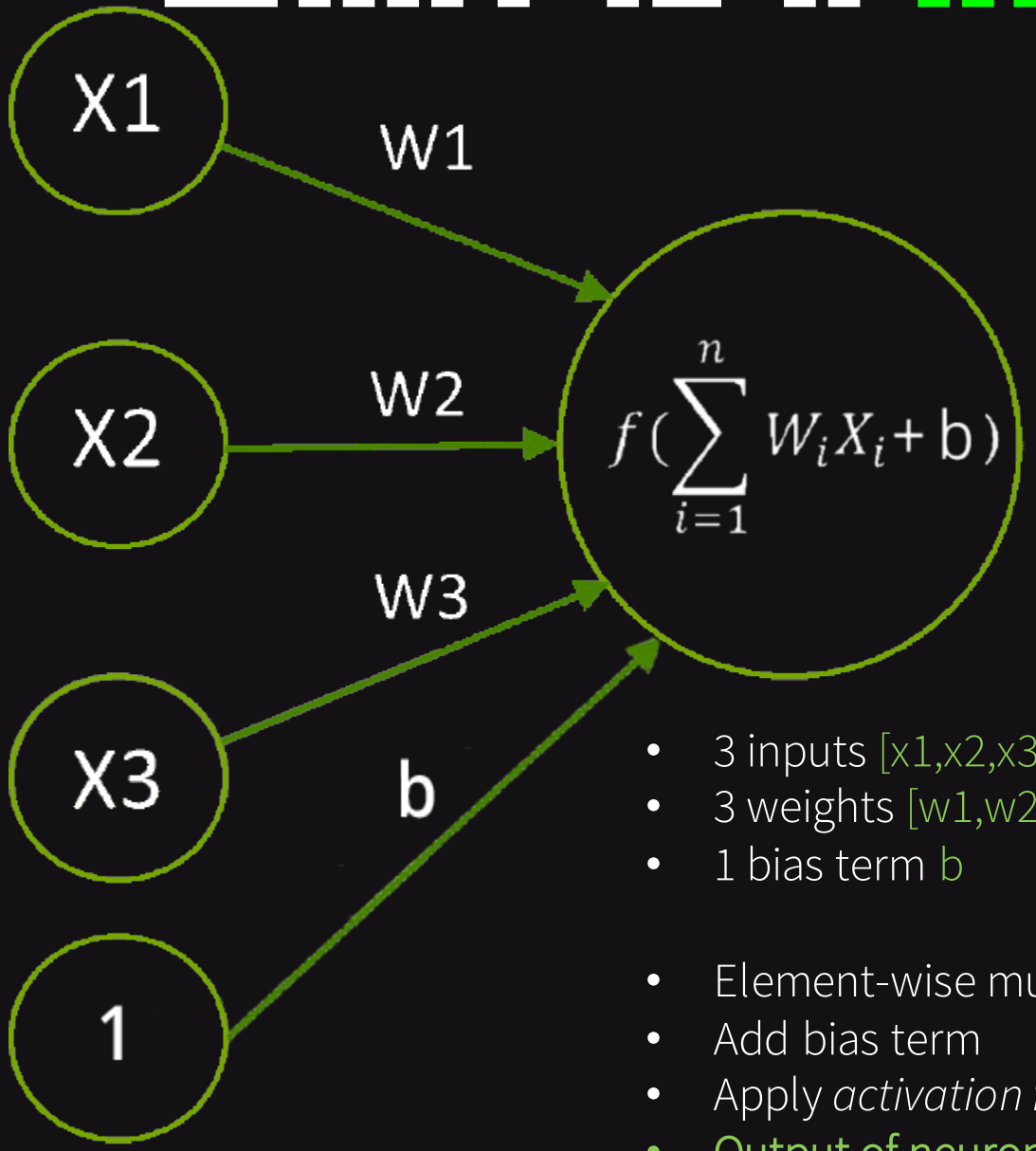
http://cs231n.github.io

Keras docs

https://keras.io/

NEURAL NETWORK CRASH COURSE

WHAT IS A NEURON?



- 3 inputs [x1,x2,x3]
- 3 weights [w1,w2,w3]
- 1 bias term b

- Element-wise multiply and sum: inputs & weights
- Add bias term
- Apply *activation function* f
- Output of neuron is just a number (like the inputs!)

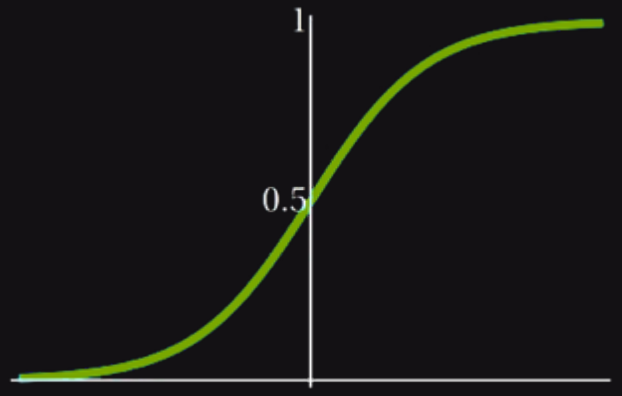
$$f\left(\sum_{i=1}^{n} W_i X_i + \mathrm{b}\right)$$

LINEAR

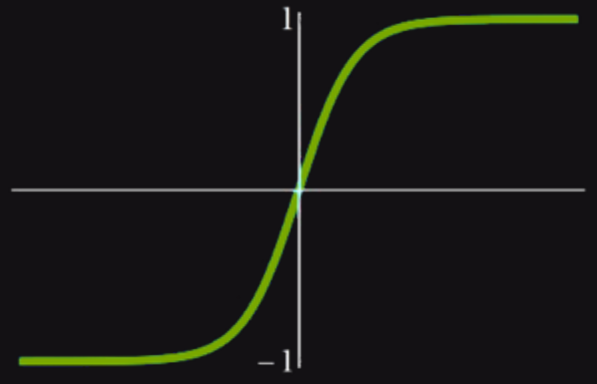
$f(\quad\quad)$

NON-LINEAR

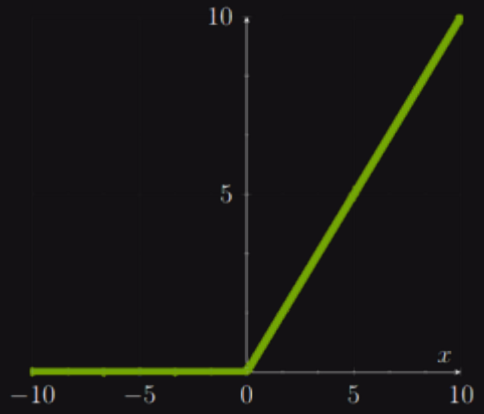
What is an Activation Function?



Sigmoid



Tanh



ReLU

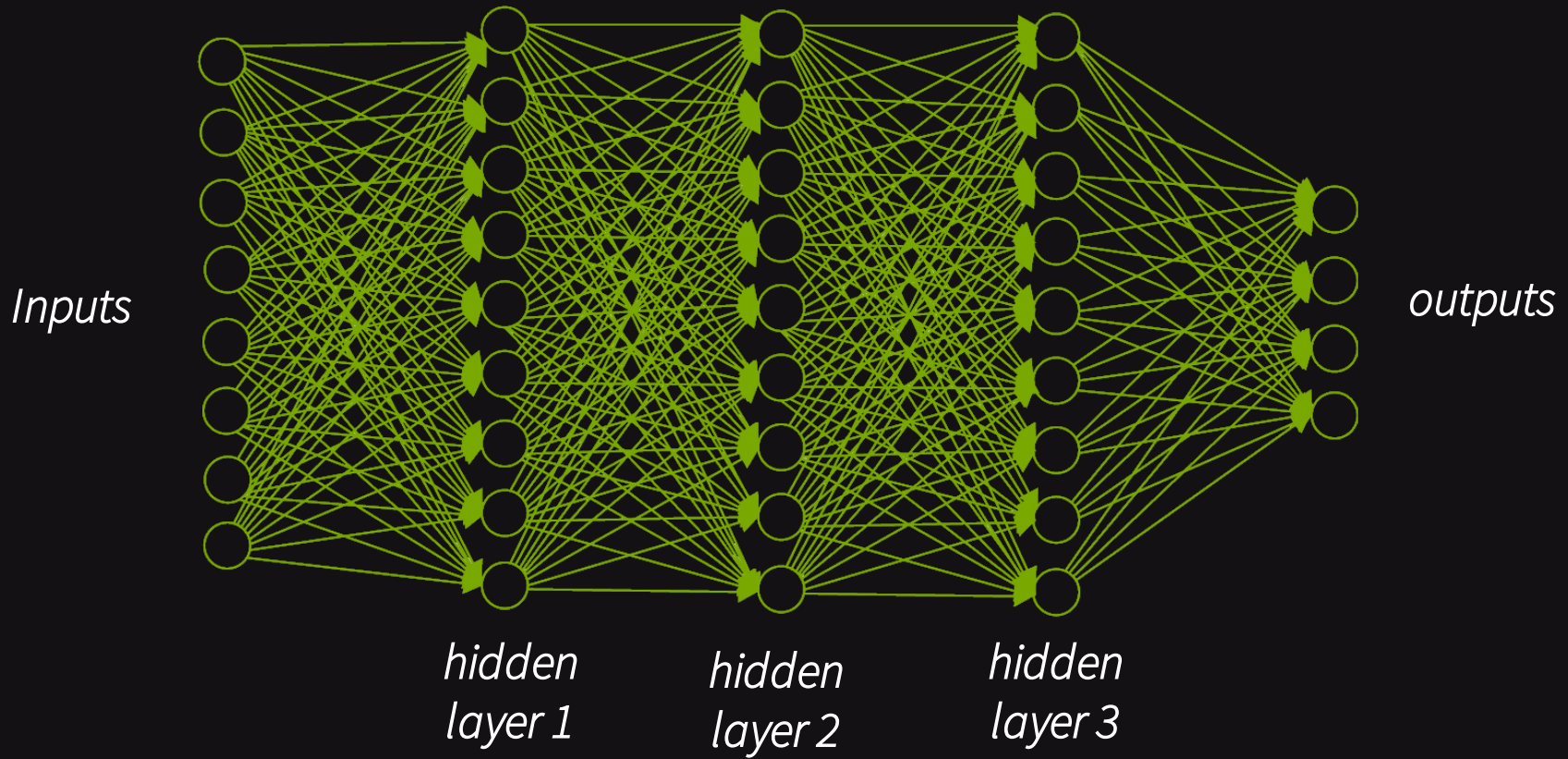
Nonlinearities … “squashing functions” … transform neuron’s output

NB: sigmoid output in [0,1]

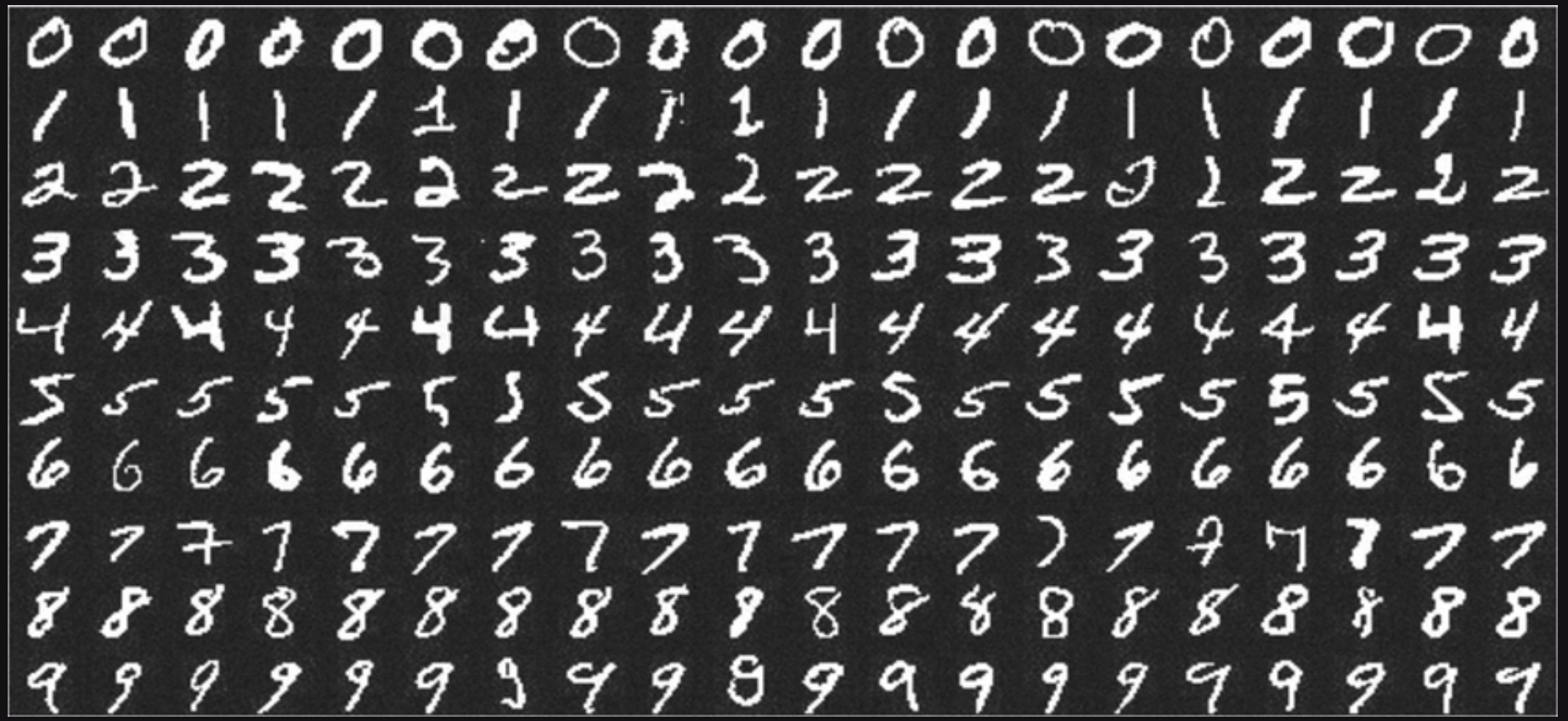
SO WHAT IS A NEURAL NETWORK?

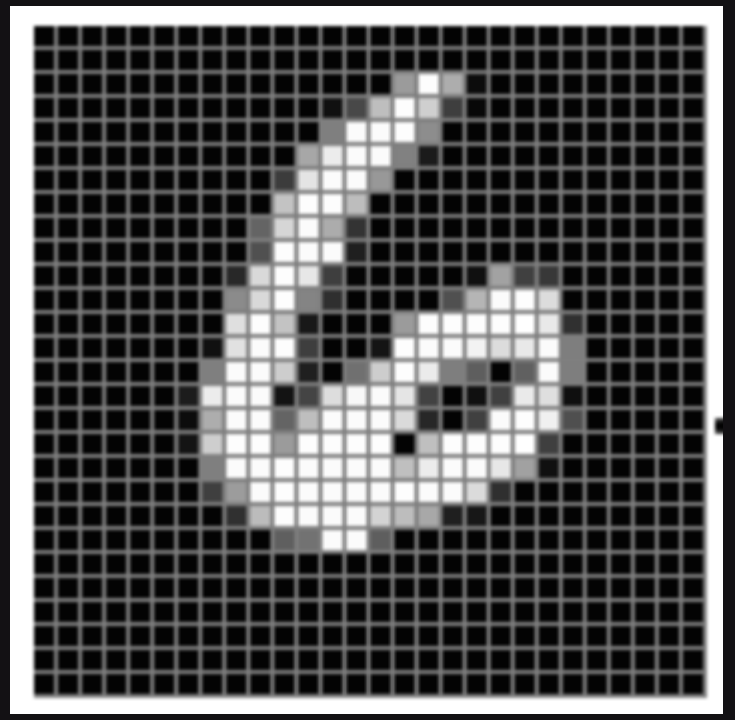
What is a (Deep) Neural Network?

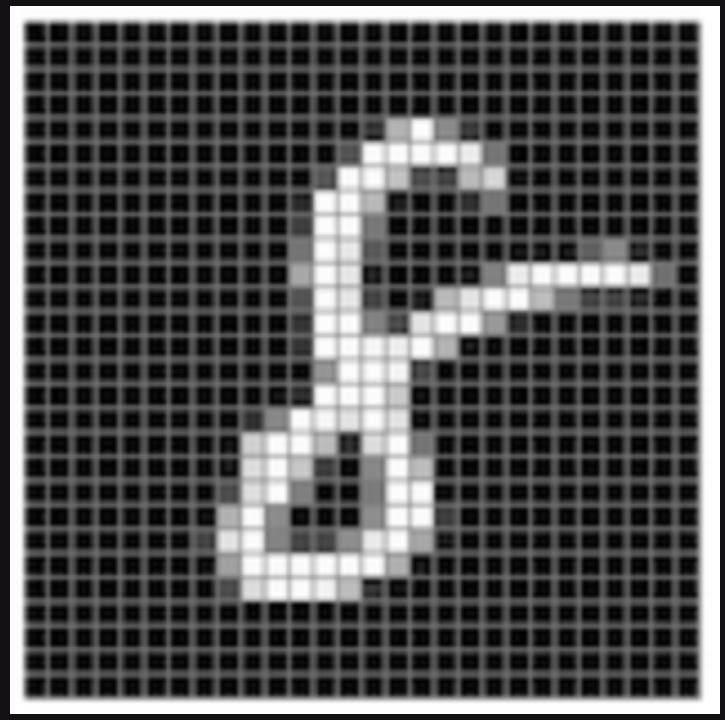
This (non-convolutional) architecture is called a "multi-layered perceptron"

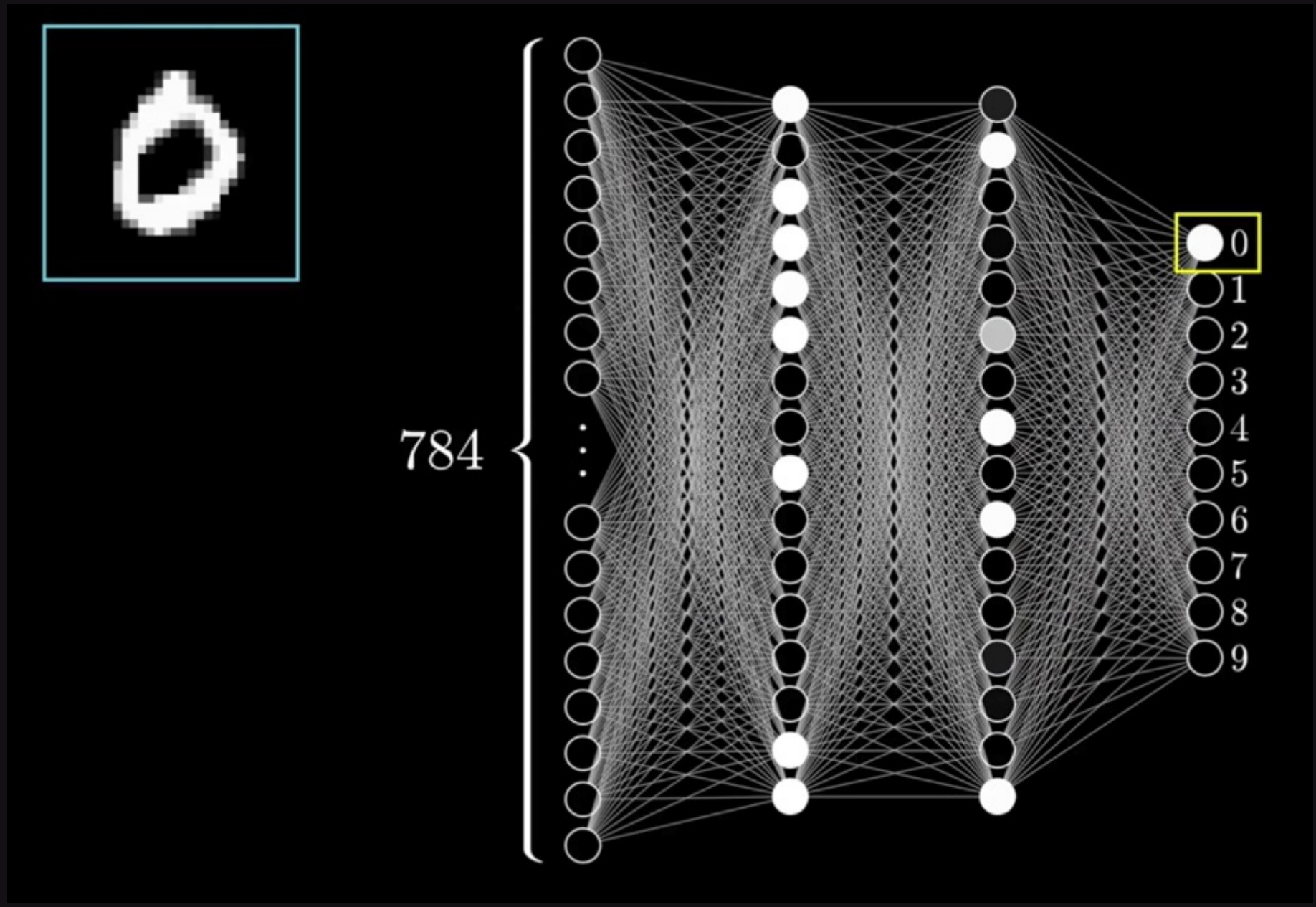


Note: Outputs of one layer are inputs into the next layer

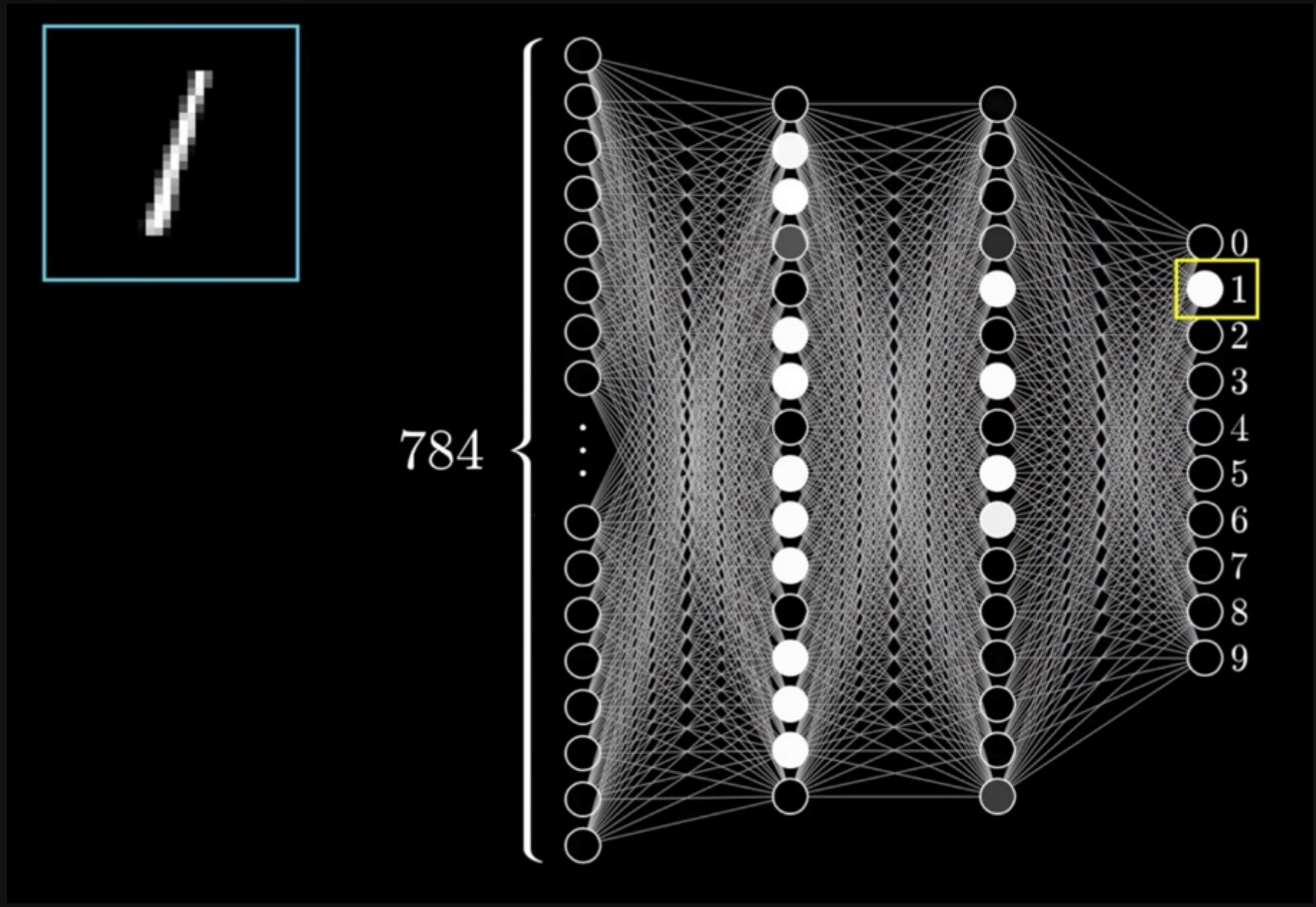




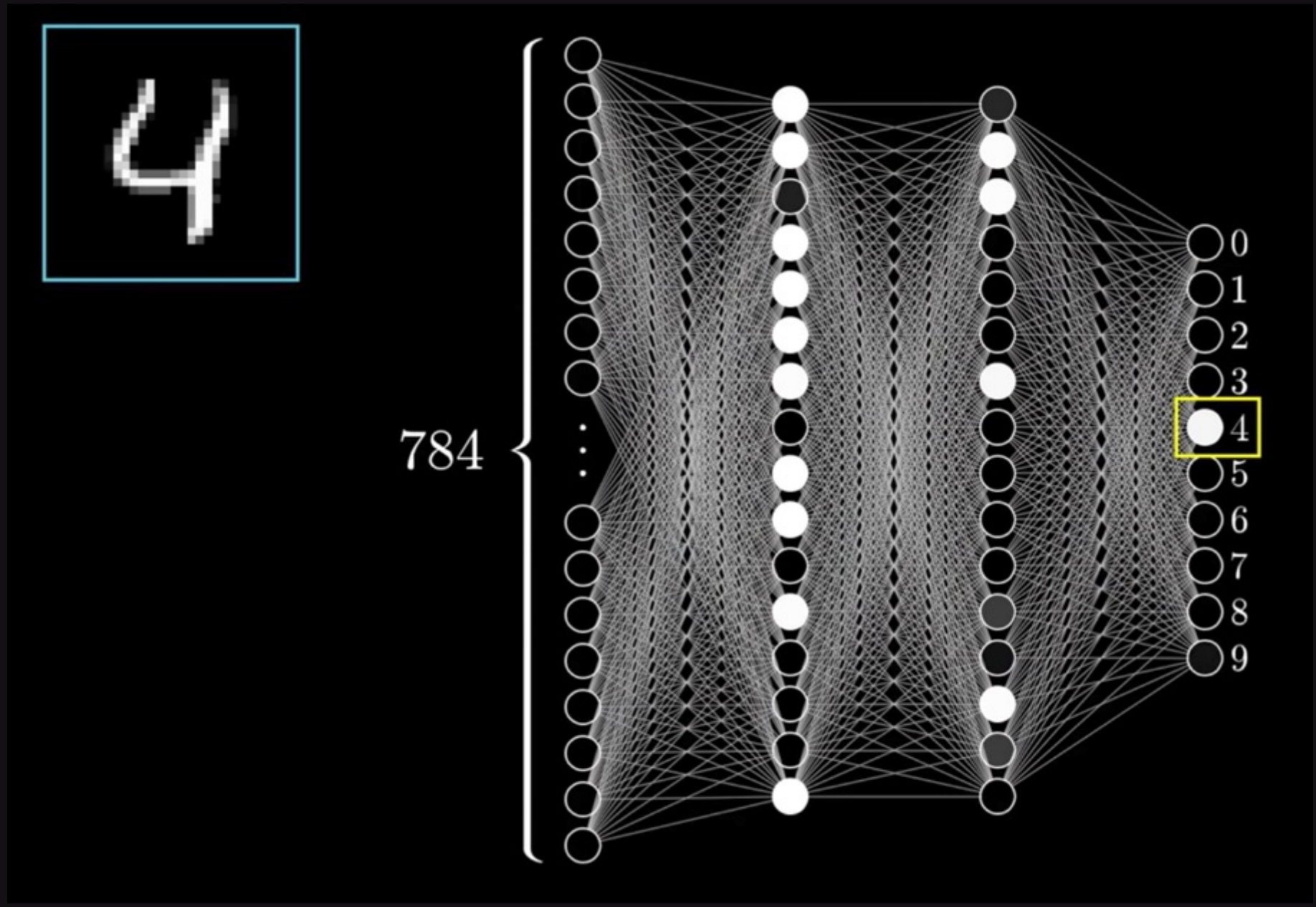




https://www.youtube.com/watch?v=aircAruvnKk

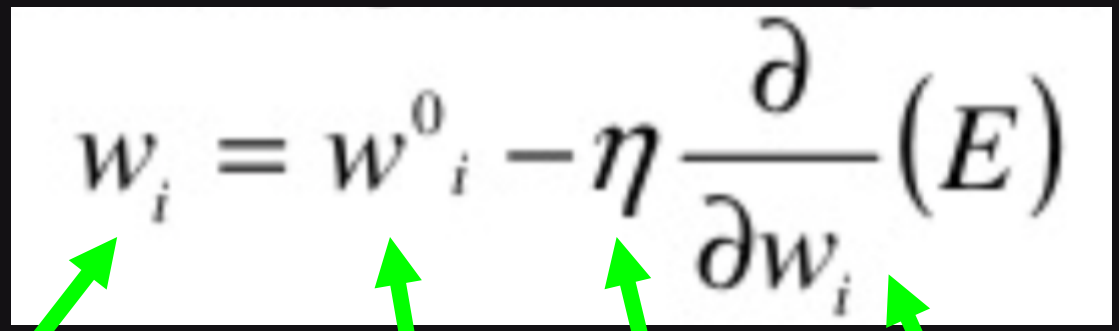


https://www.youtube.com/watch?v=aircAruvnKk

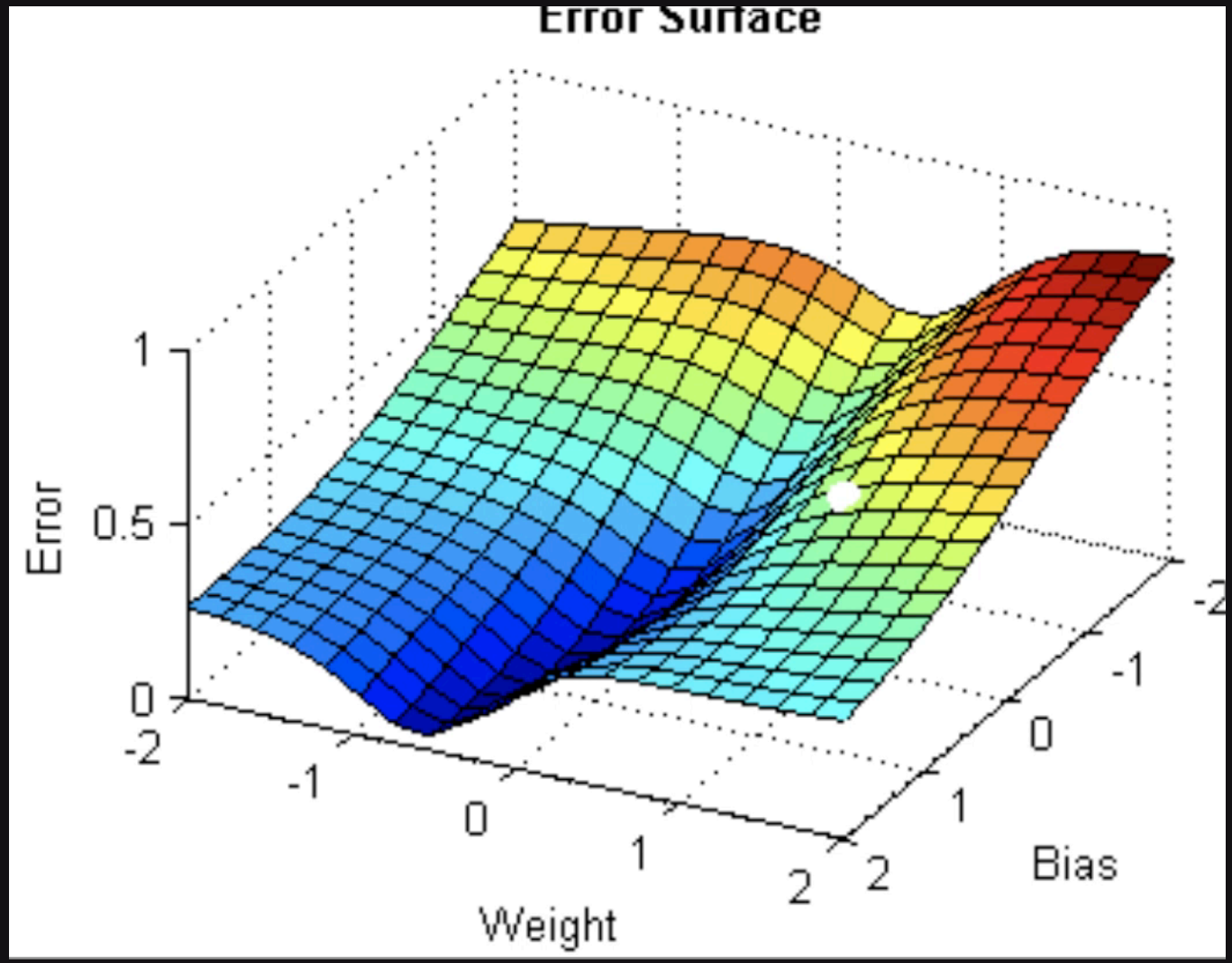


https://www.youtube.com/watch?v=aircAruvnKk

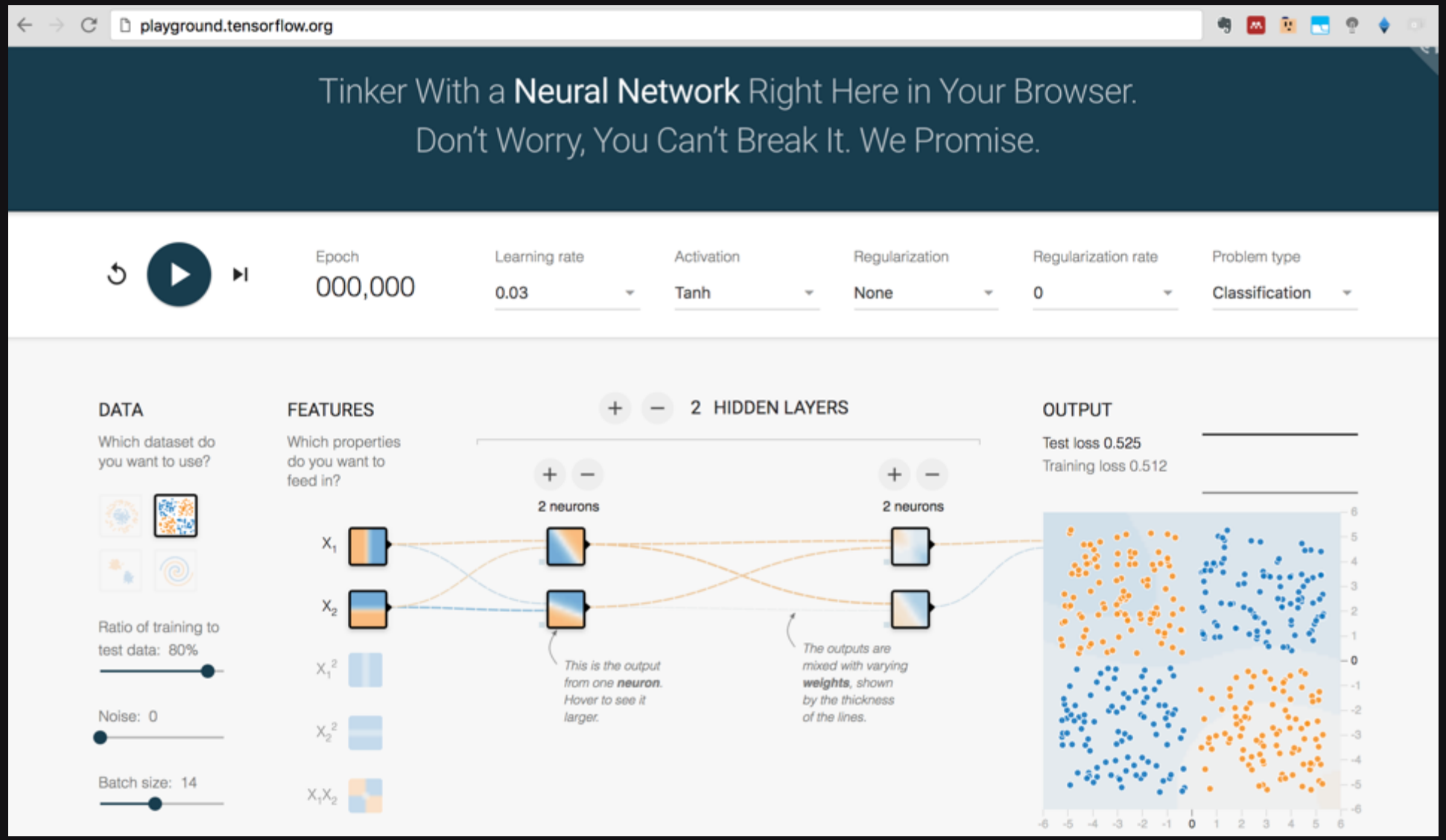
How does a neural network learn?



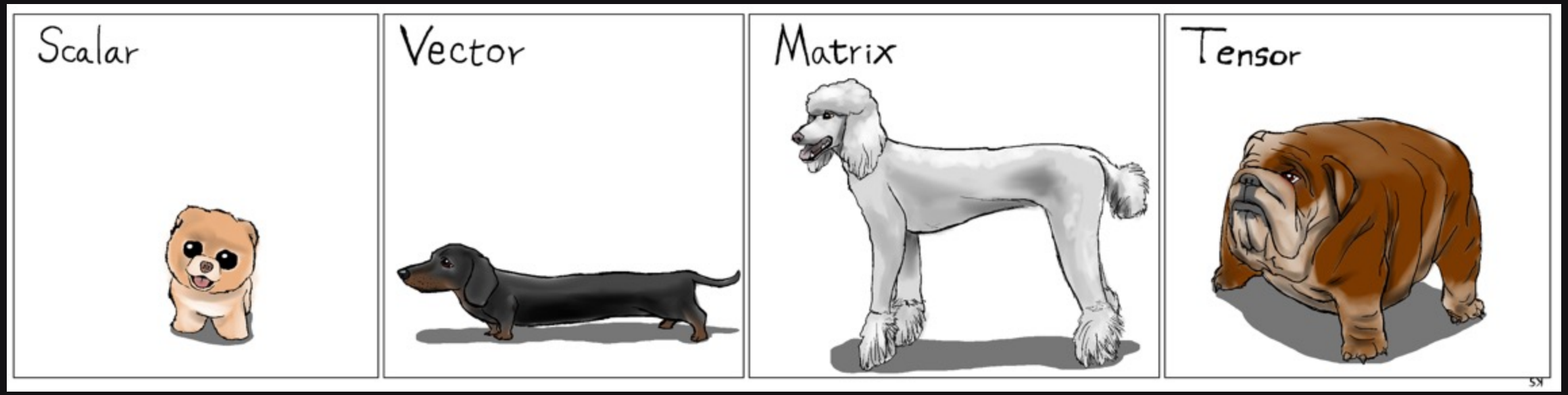
GRADIENT DESCENT



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



http://playground.tensorflow.org


Scalar
Vector
Matrix
Tensor

CONVOLUTIONAL
NEURAL NETWORK

CNN

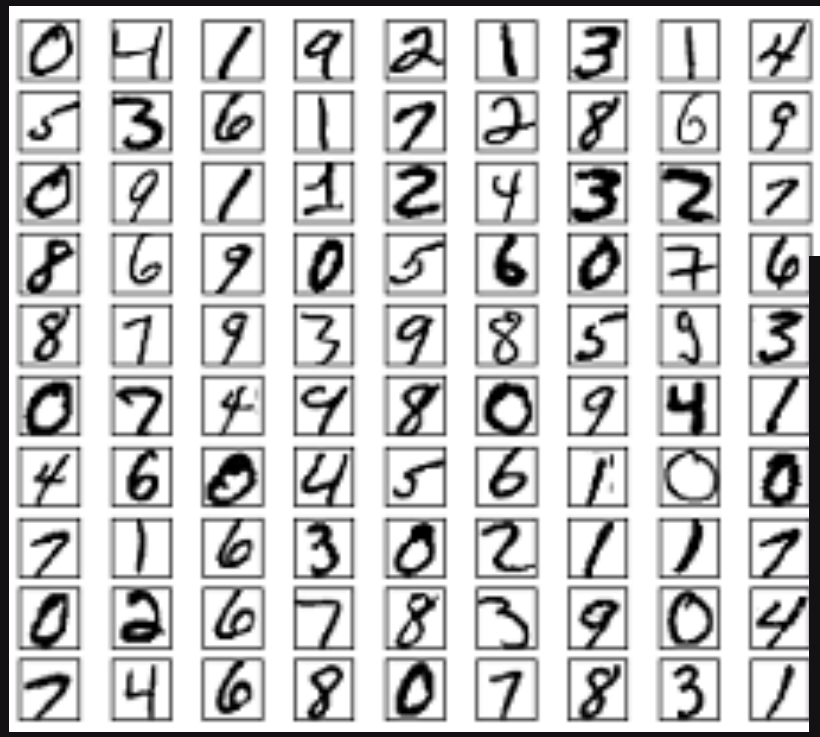
IMAGE CLASSIFICATION

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

IMAGE & VIDEO MODERATION

Large international gay dating app with tens of millions of users uploading hundreds-of-thousands of photos per day

IMAGE CLASSIFICATION

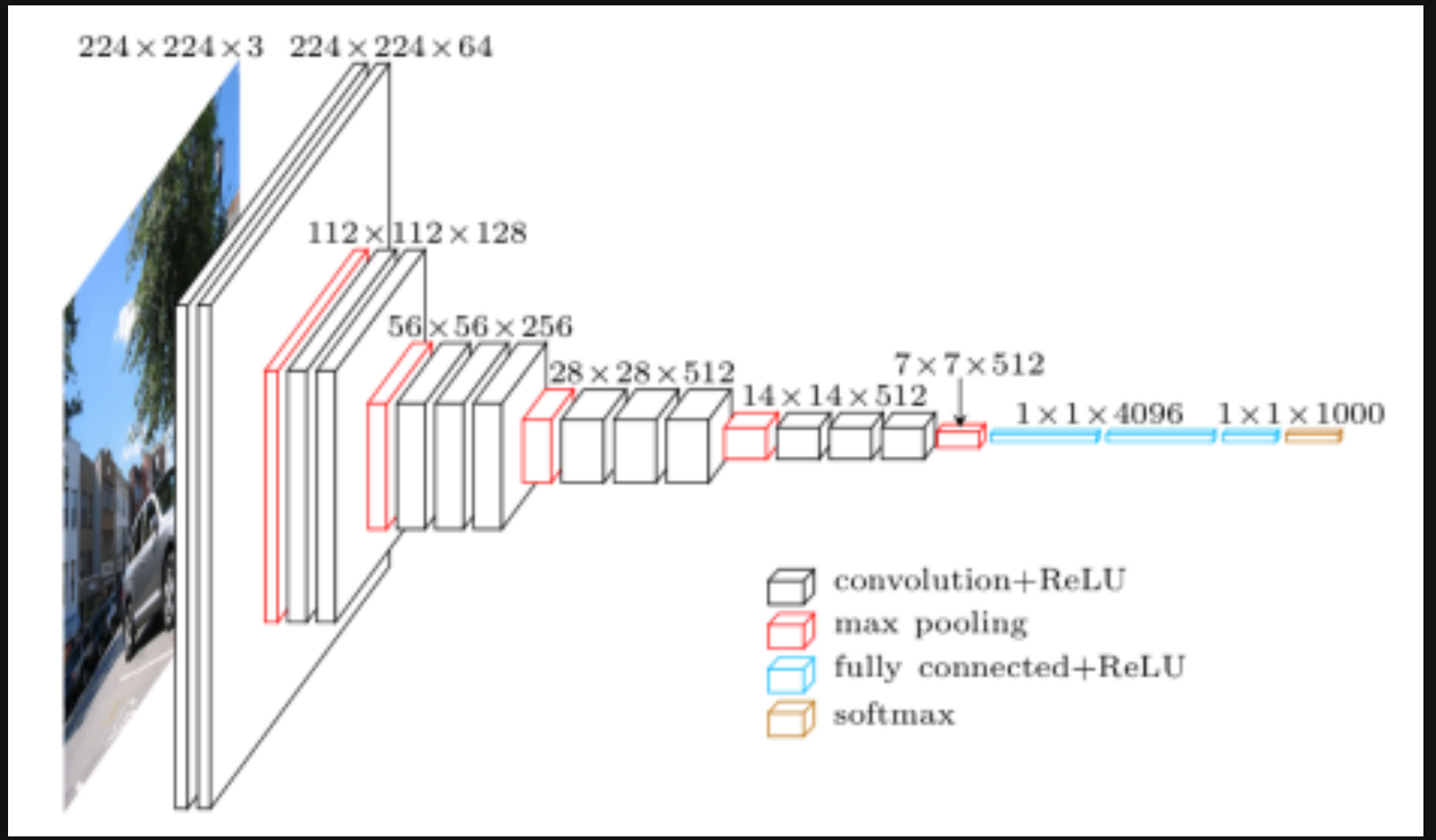


http://yann.lecun.com/exdb/mnist/

https://github.com/fchollet/keras/blob/master/examples/mnist_cnn.py
(99.25% test accuracy in 192 seconds and 46 lines of code)

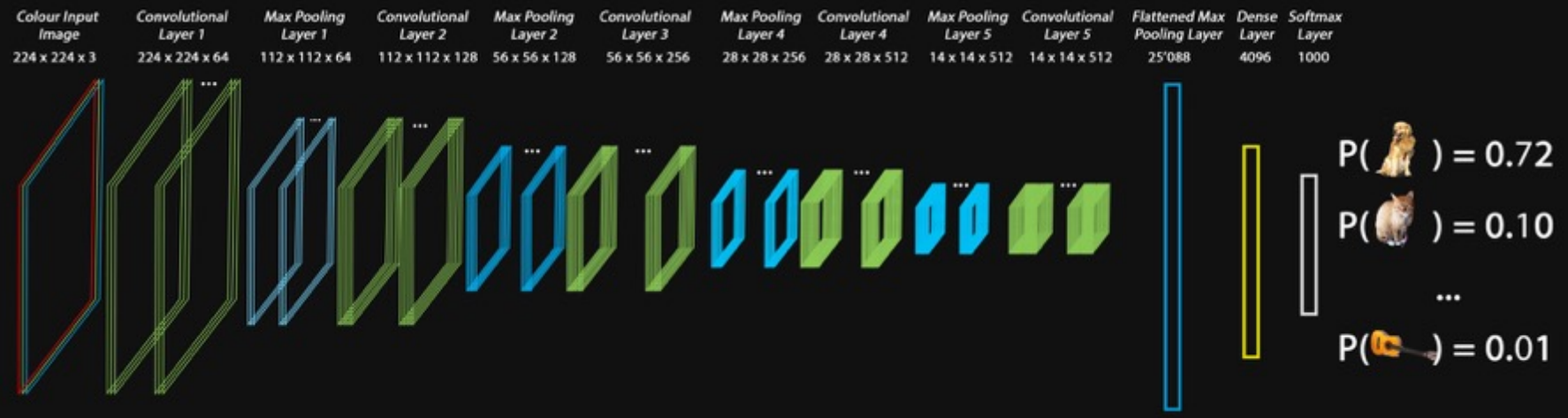


EXAMPLE ARCHITECTURE



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

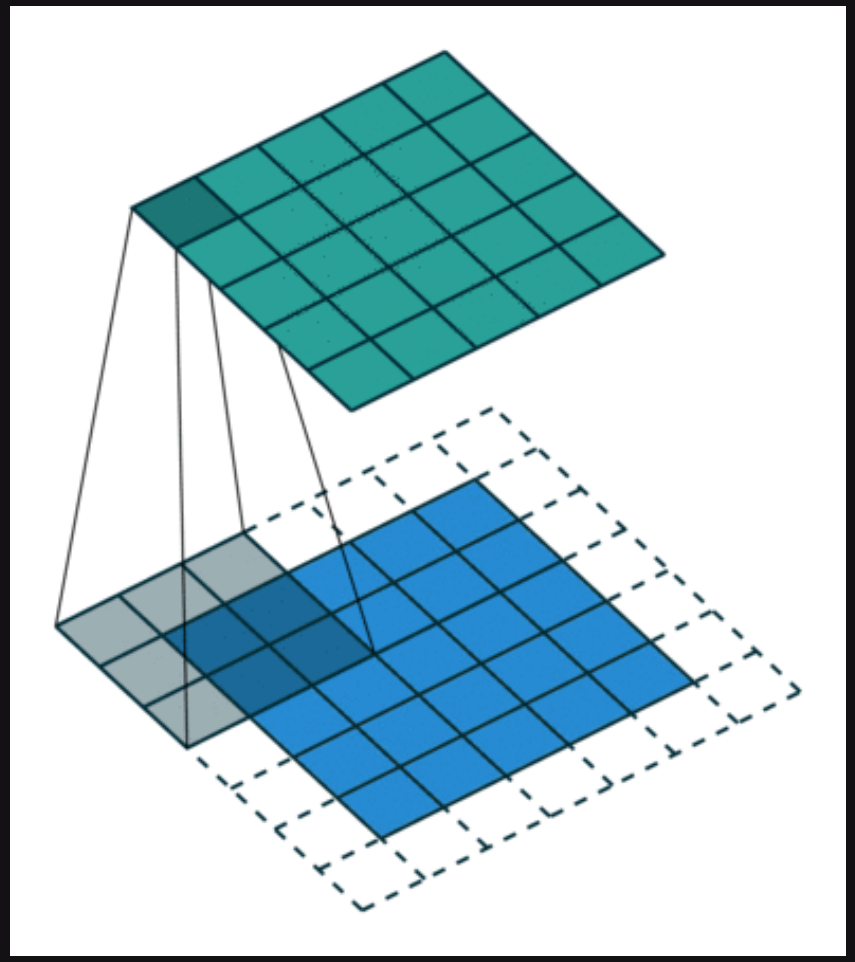
EXAMPLE ARCHITECTURE



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

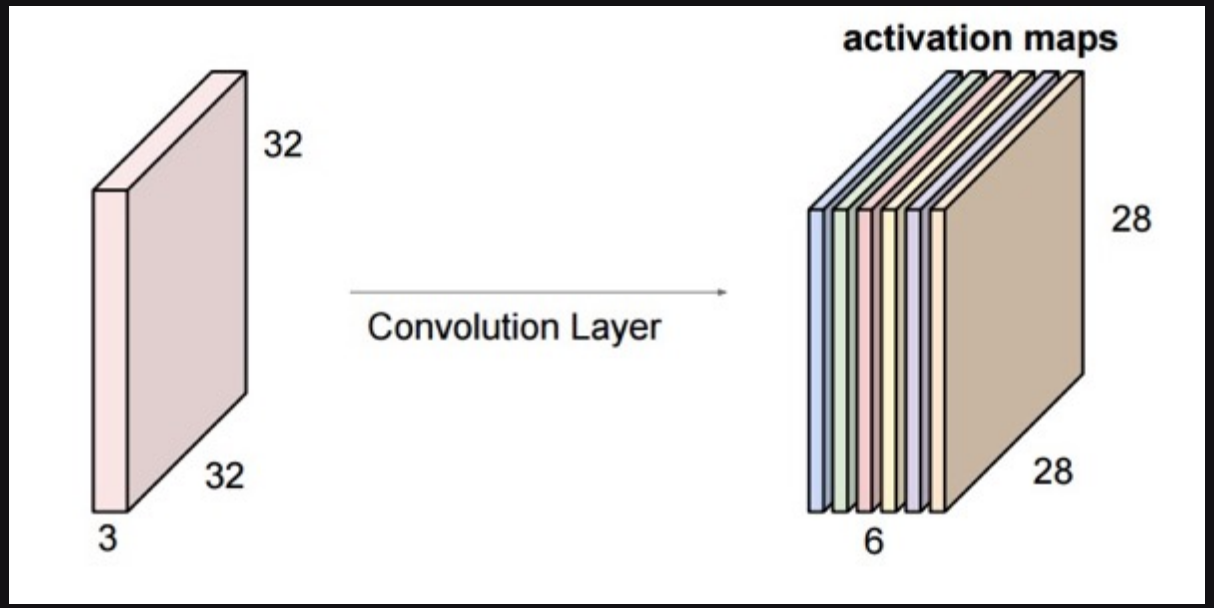
CONVOLUTIONS

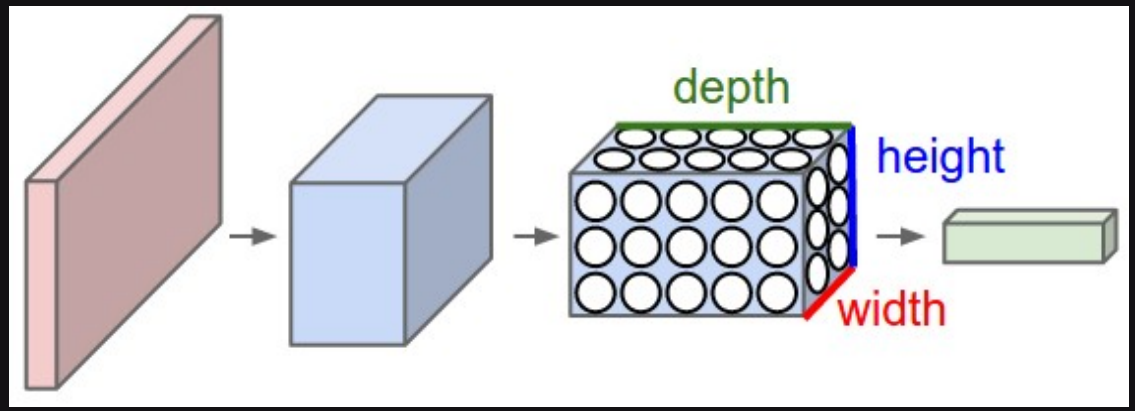
http://setosa.io/ev/image-kernels/



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

MANY KERNELS = MANY ACTIVATION MAPS = VOLUME





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

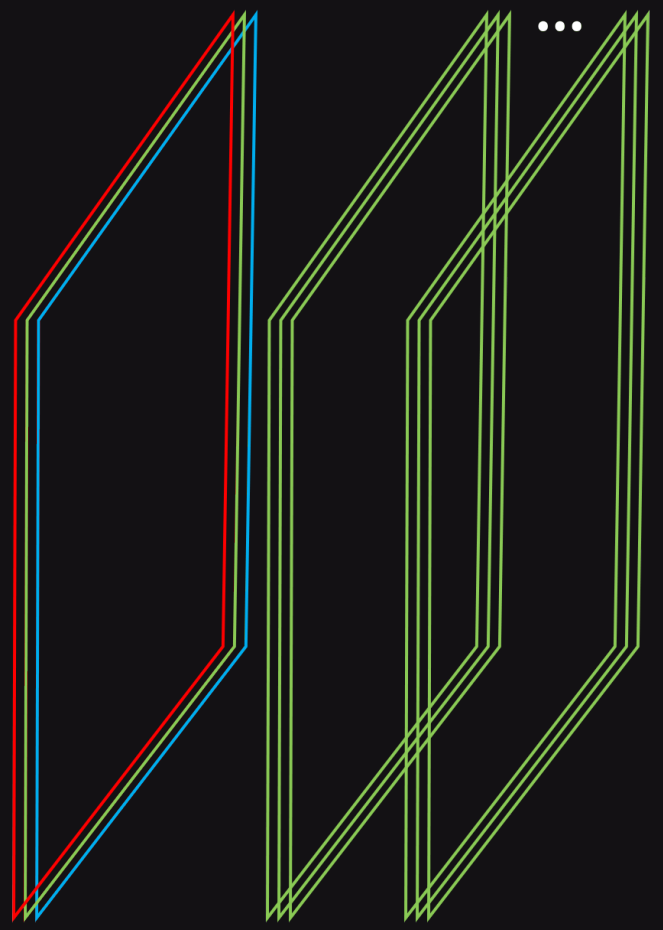
CONVOLUTIONAL LAYER

Colour Input Image

224 x 224 x 3

Convolutional Layer 1

224 x 224 x 64



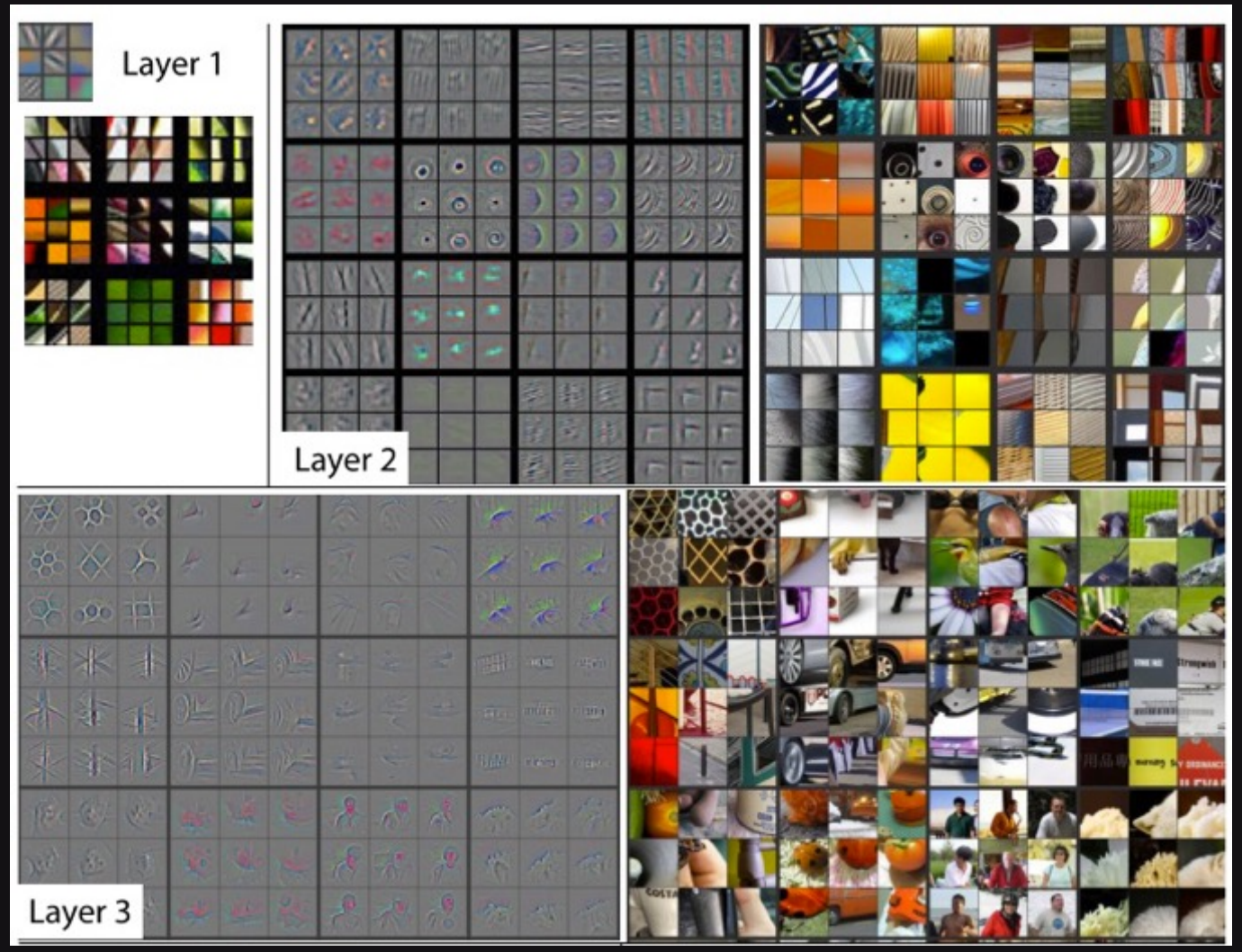
Visualizing and Understanding Convolutional Networks

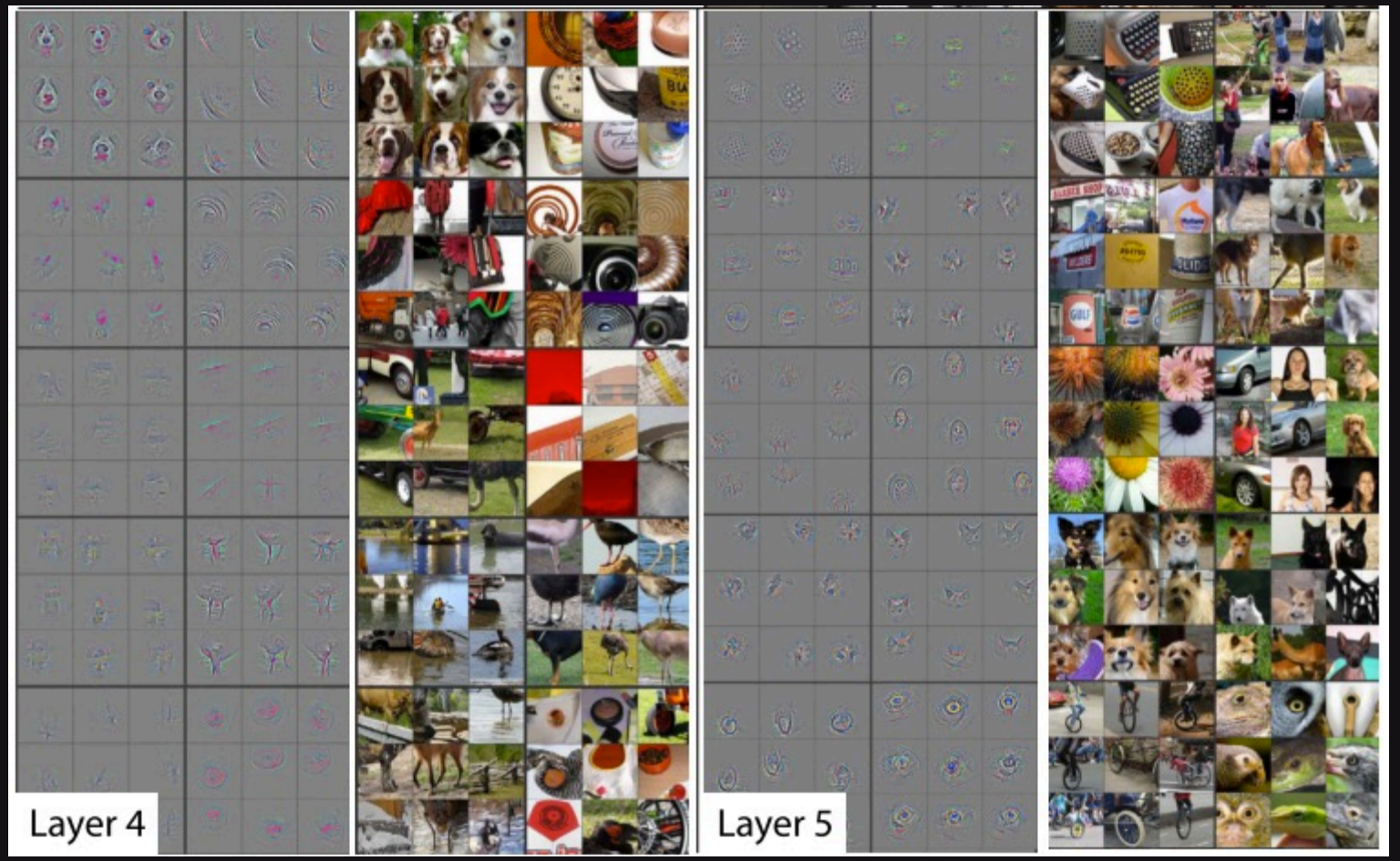
Matthew D. Zeiler ZEILER@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University

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

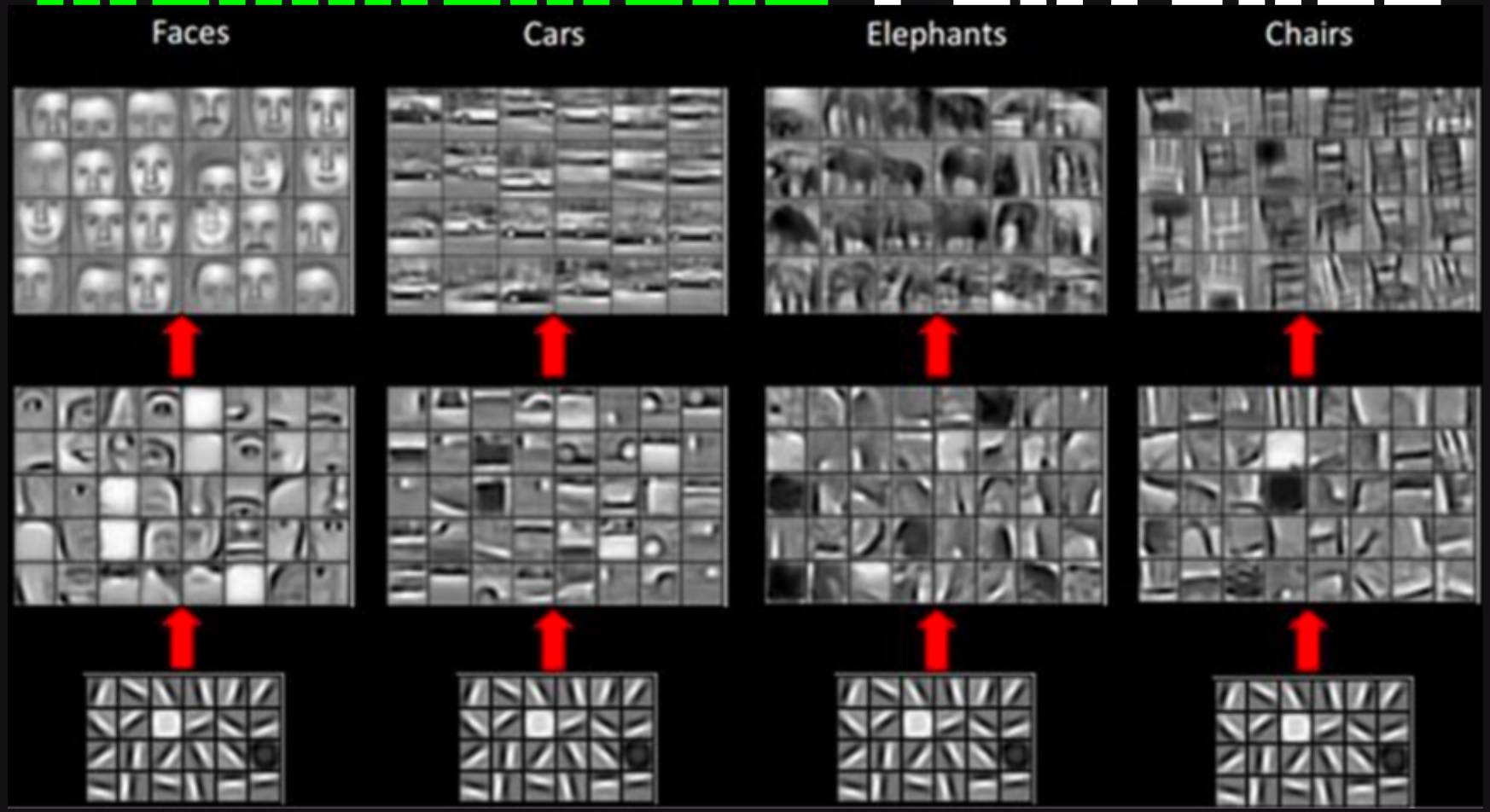
https://github.com/fchollet/keras/blob/master/examples/conv_filter_visualization.py


Layer 1

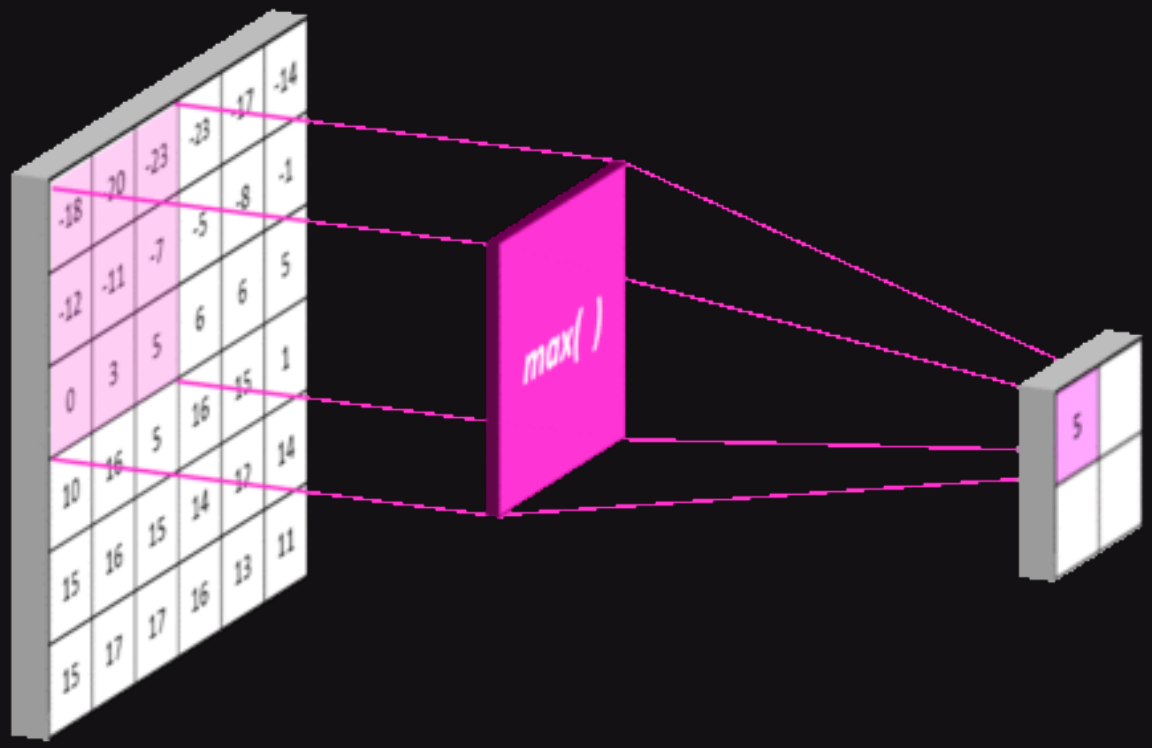

Layer 1
Layer 2
Layer 3


Layer 4
Layer 5

CONVOLUTION LEARN HIERARCHICAL FEATURES

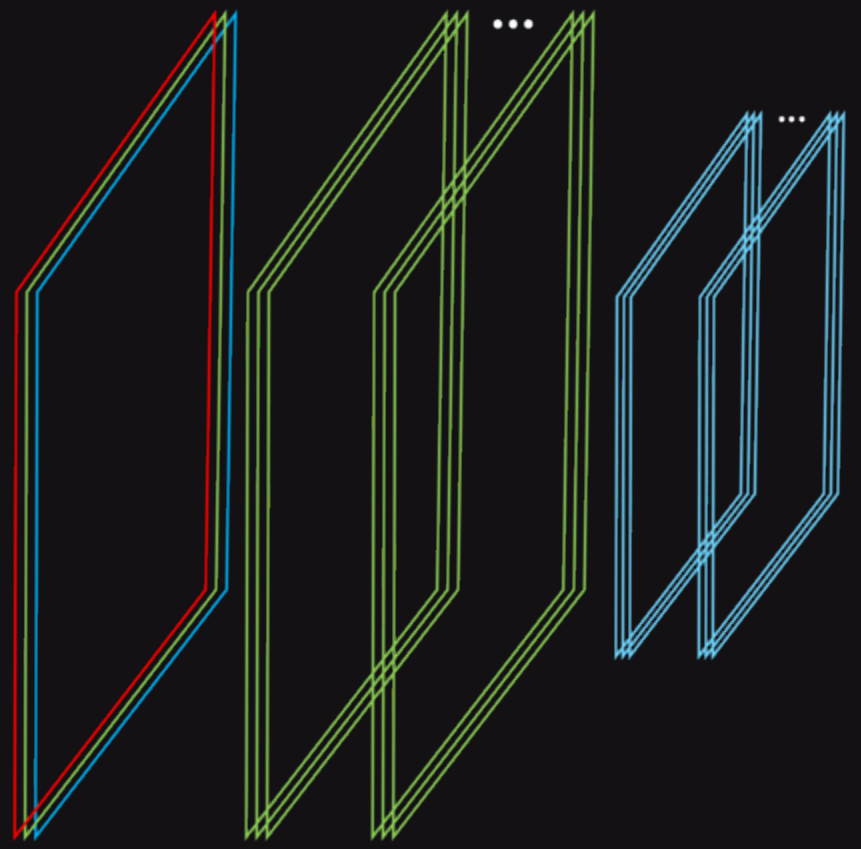


MAX POOLING


-14
-17
-23
-23
-1
-8
-16
20
-5
-7
-11
-12
5
6
6
5
3
0
1
15
16
5
14
17
10
11
13
16
17
15
15
17
max( )
5

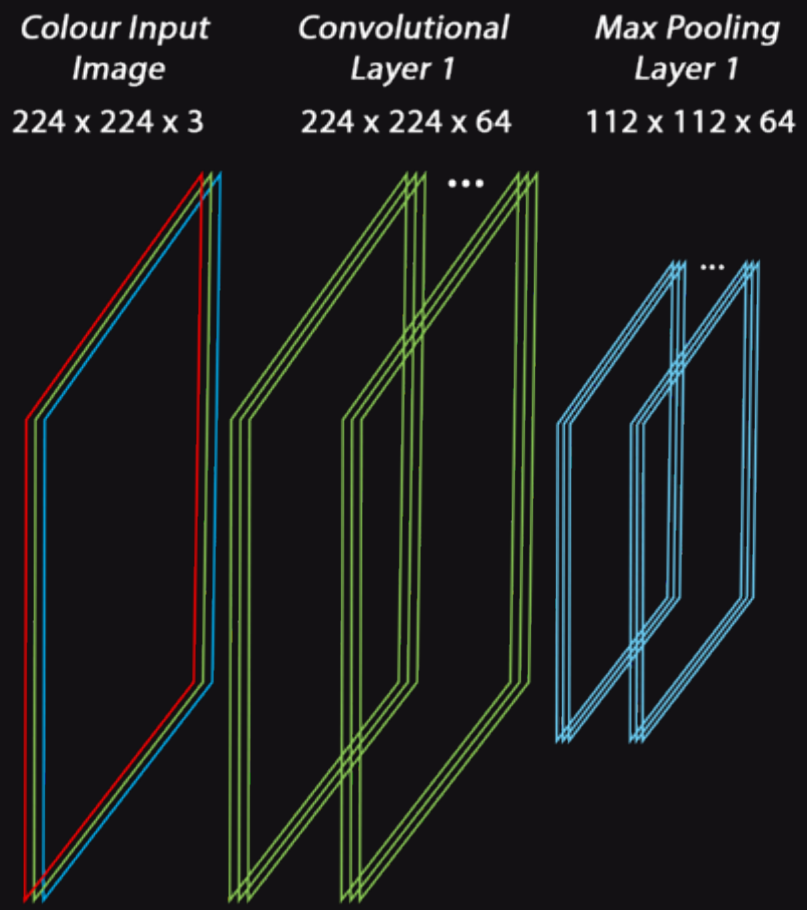
MAX POOLING

Colour Input Image	*Convolutional Layer 1*	*Max Pooling Layer 1*
224 x 224 x 3	224 x 224 x 64	112 x 112 x 64

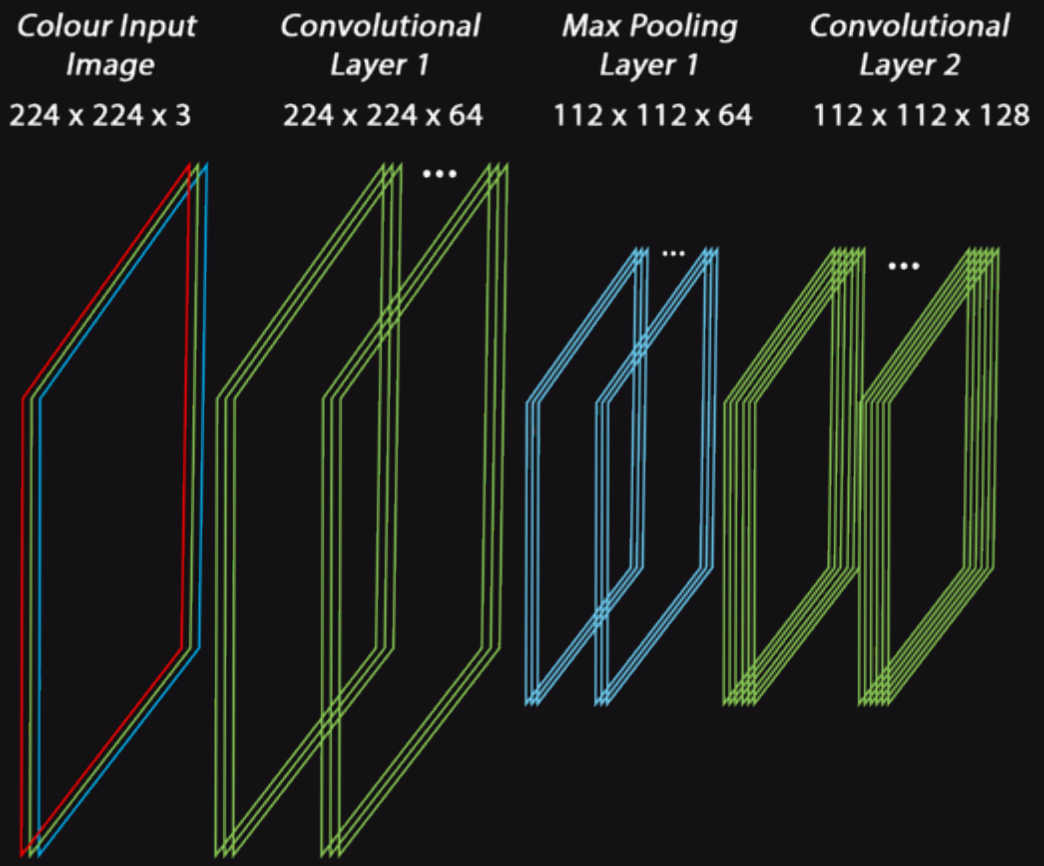


RINSE WASH REPEAT ... DEEEEEP!

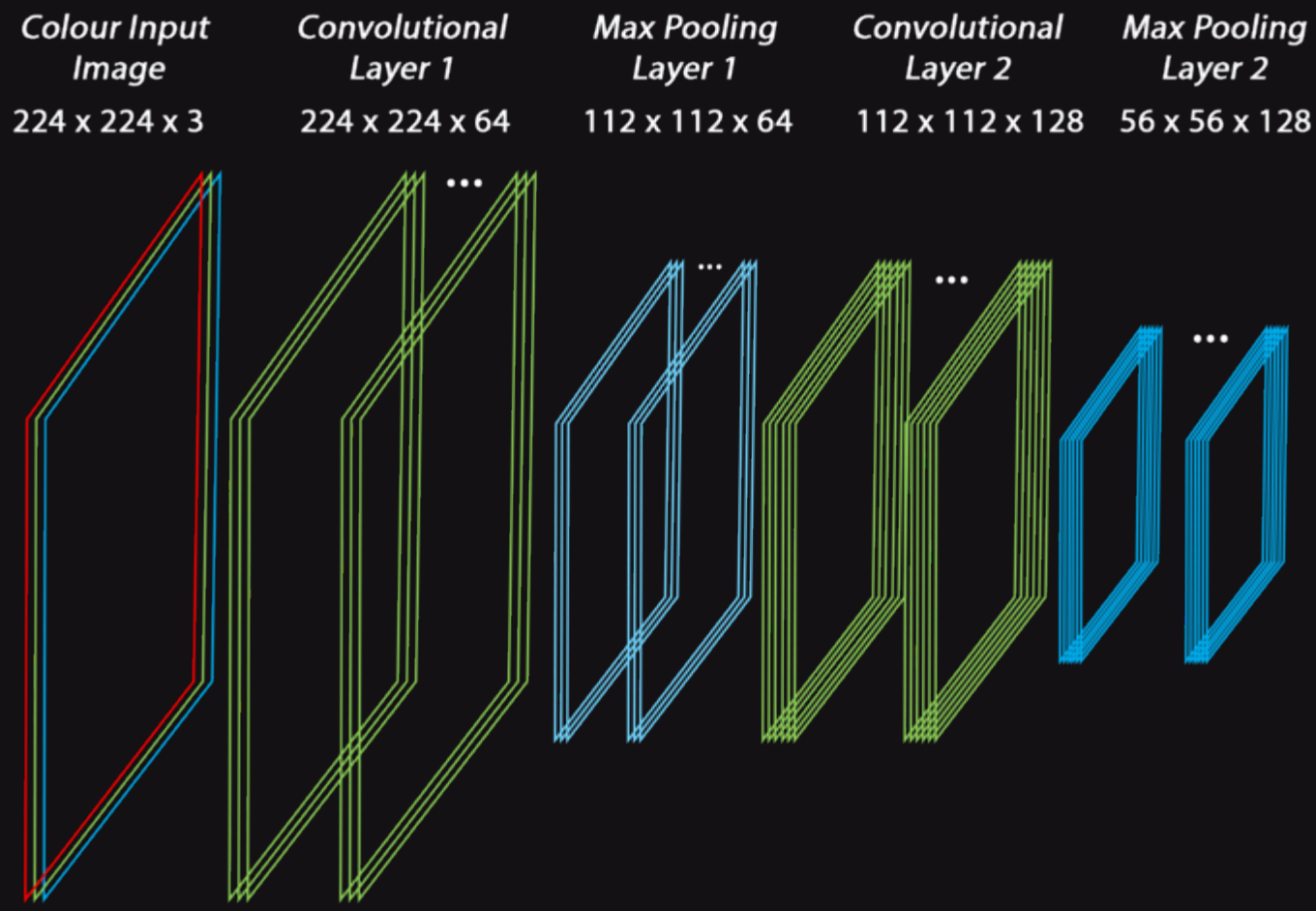
Convolution + max pooling



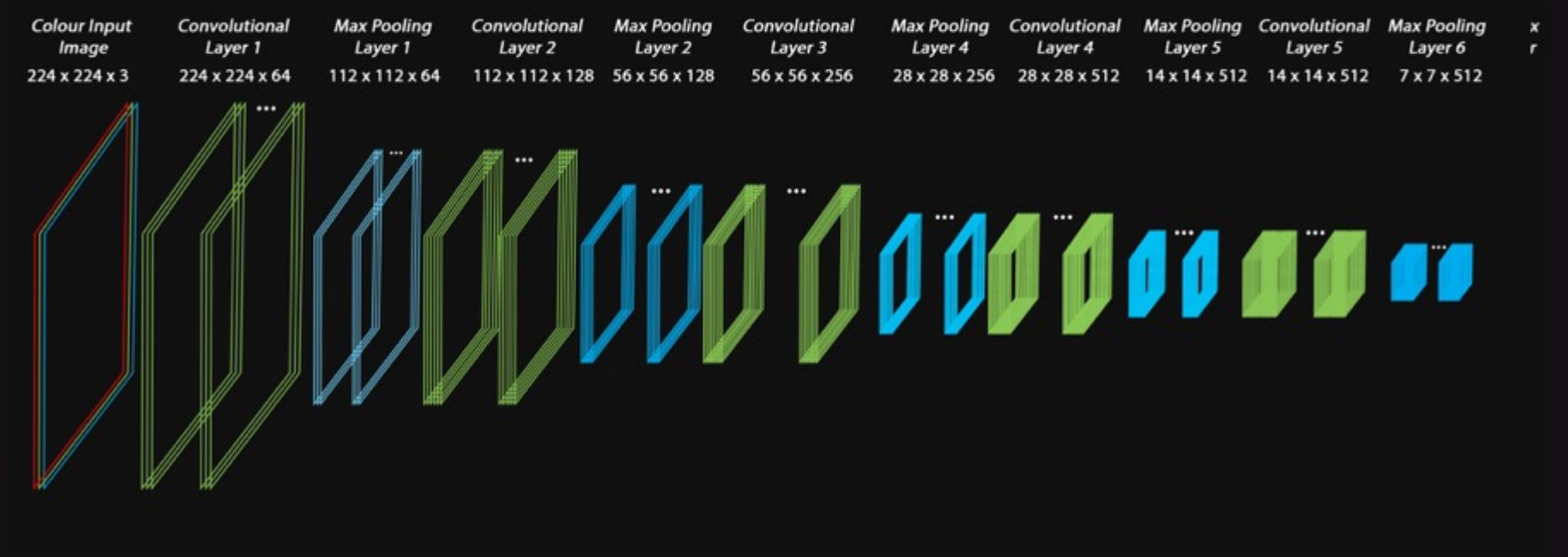
Convolution + max pooling



STACK LAYERS



STACK LAYERS



FLATTEN

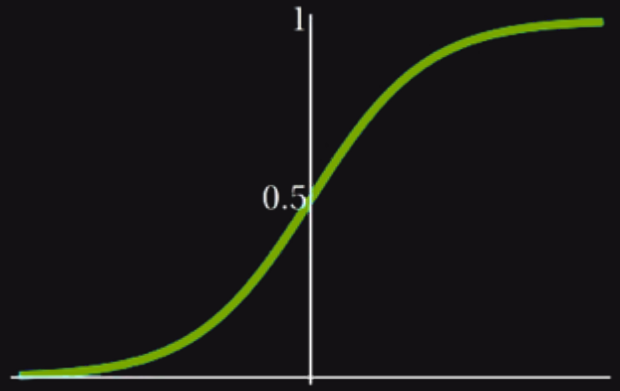
Convolution + max pooling

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

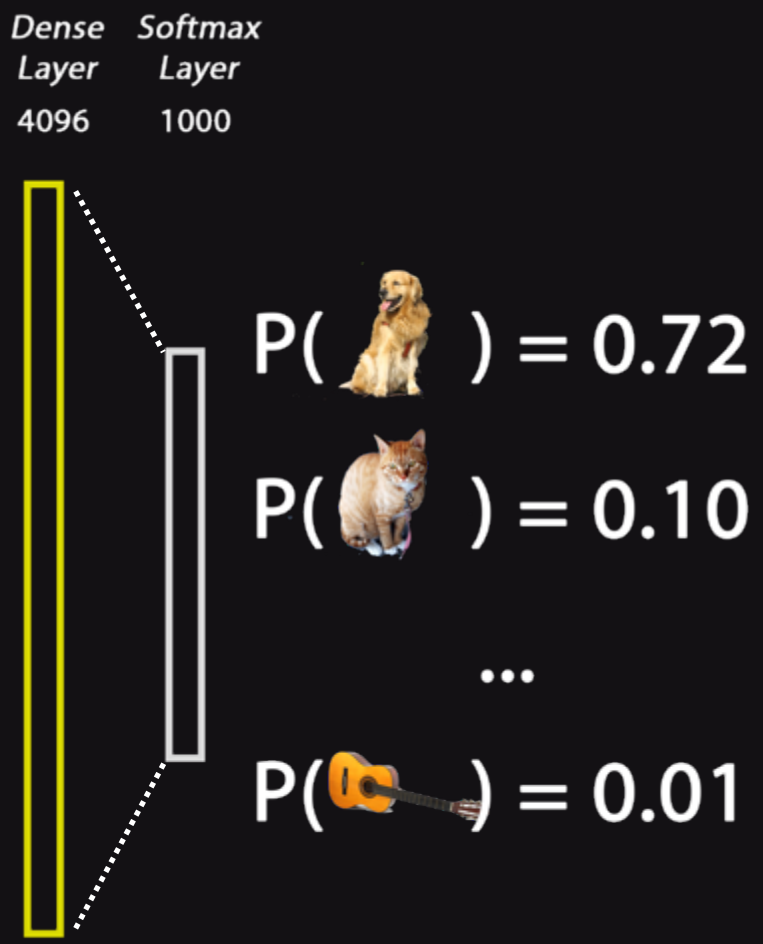
SOFTMAX

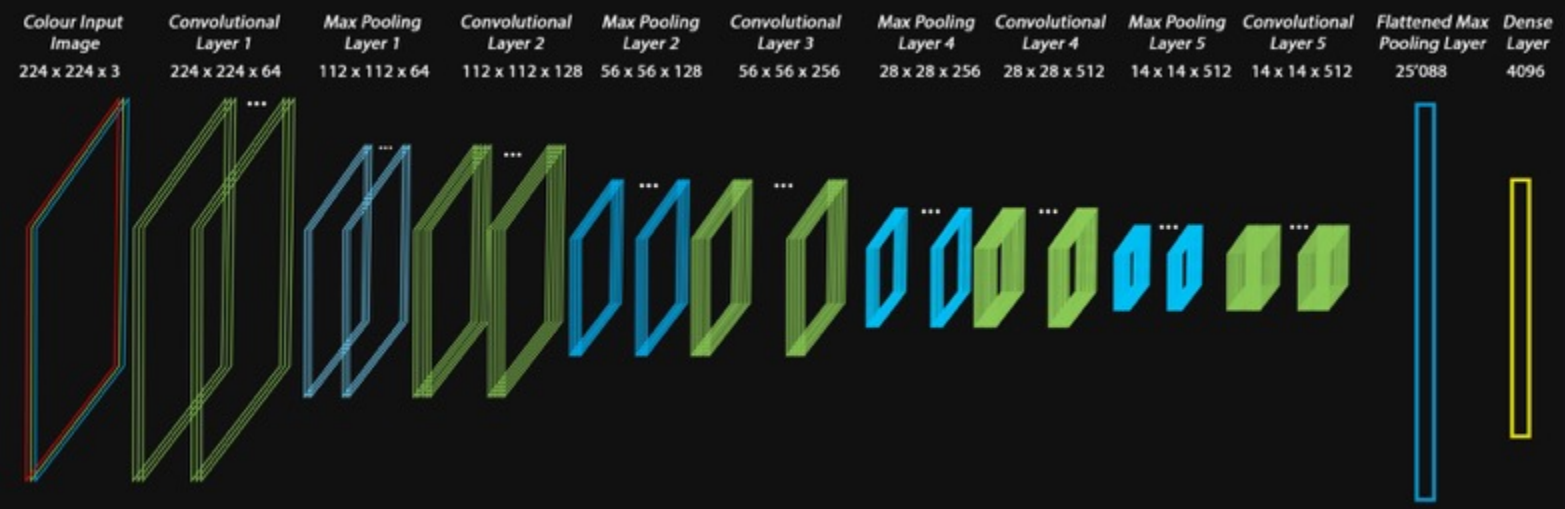


Convert scores $\in \mathbb{R}$ to probabilities $\in [0,1]$

Final output prediction = highest probability class

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




Colour Input Image
224 x 224 x 3
Convolutional Layer 1
224 x 224 x 64
Max Pooling Layer 1
112 x 112 x 64
Convolutional Layer 2
112 x 112 x 128
Max Pooling Layer 2
56 x 56 x 128
Convolutional Layer 3
56 x 56 x 256
Max Pooling Layer 4
28 x 28 x 256
Convolutional Layer 4
28 x 28 x 512
Max Pooling Layer 5
14 x 14 x 512
Convolutional Layer 5
14 x 14 x 512
Flattened Max Pooling Layer
25'088
Dense Layer
4096

WE NEED LABELLED

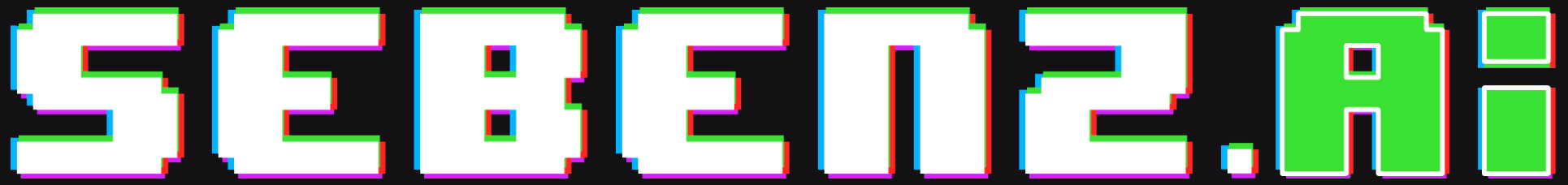
TRAINING DATA!

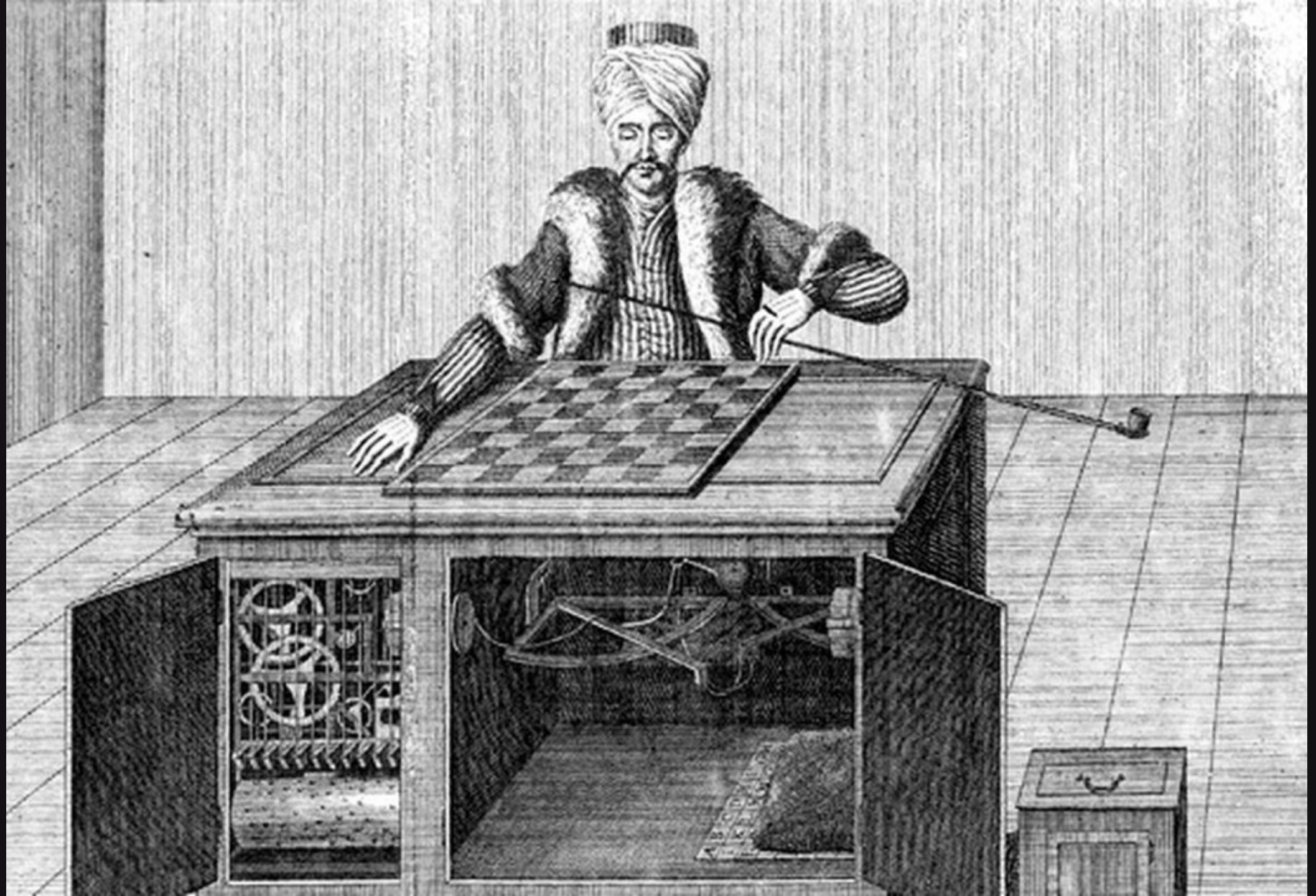
IMAGENET

1000 object categories

1.2 million training images

http://image-net.org/explore


SEBEN2.AI



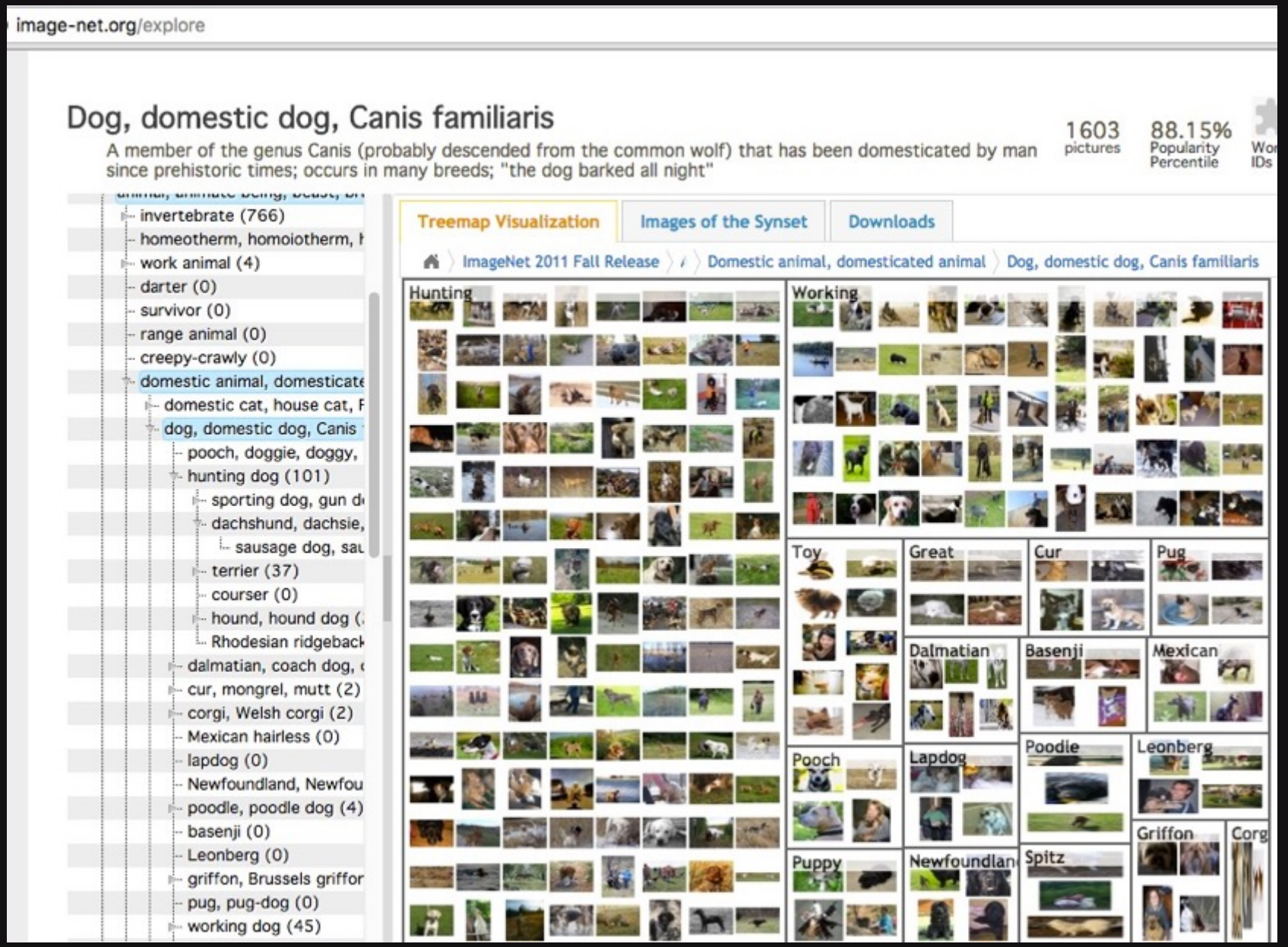


image-net.org/explore
Dog, domestic dog, Canis familiaris
A member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds; "the dog barked all night"
1603 pictures
88.15% Popularity Percentile
Treemap Visualization
Images of the Synset
Downloads
ImageNet 2011 Fall Release
Domestic animal, domesticated animal
Dog, domestic dog, Canis familiaris
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)
Hunting
Working
Toy
Great
Cur
Pug
Dalmatian
Basenji
Mexican
Pooch
Lapdog
Poodle
Leonberg
Puppy
Newfoundlan
Spitz
Griffon
Corg

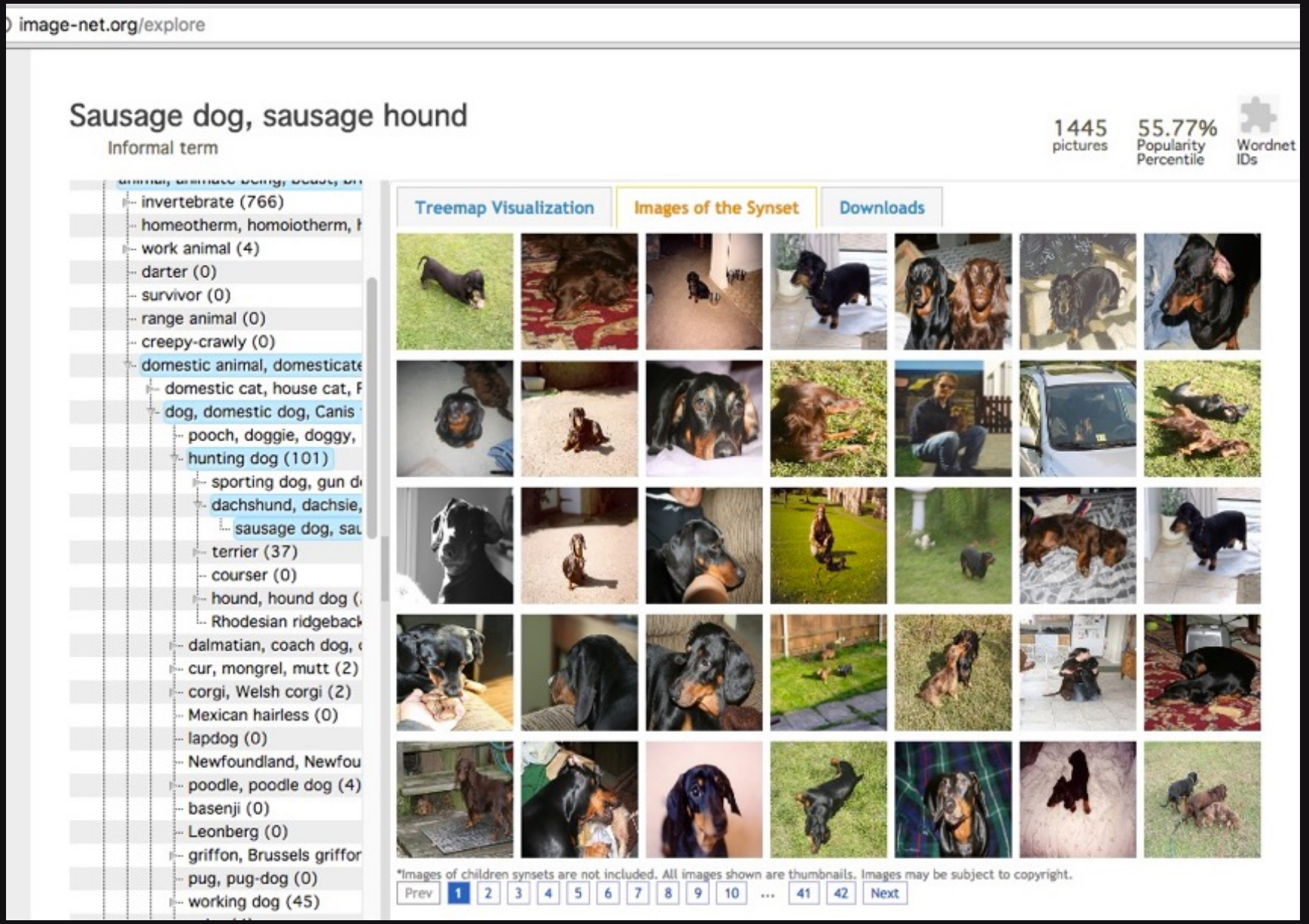
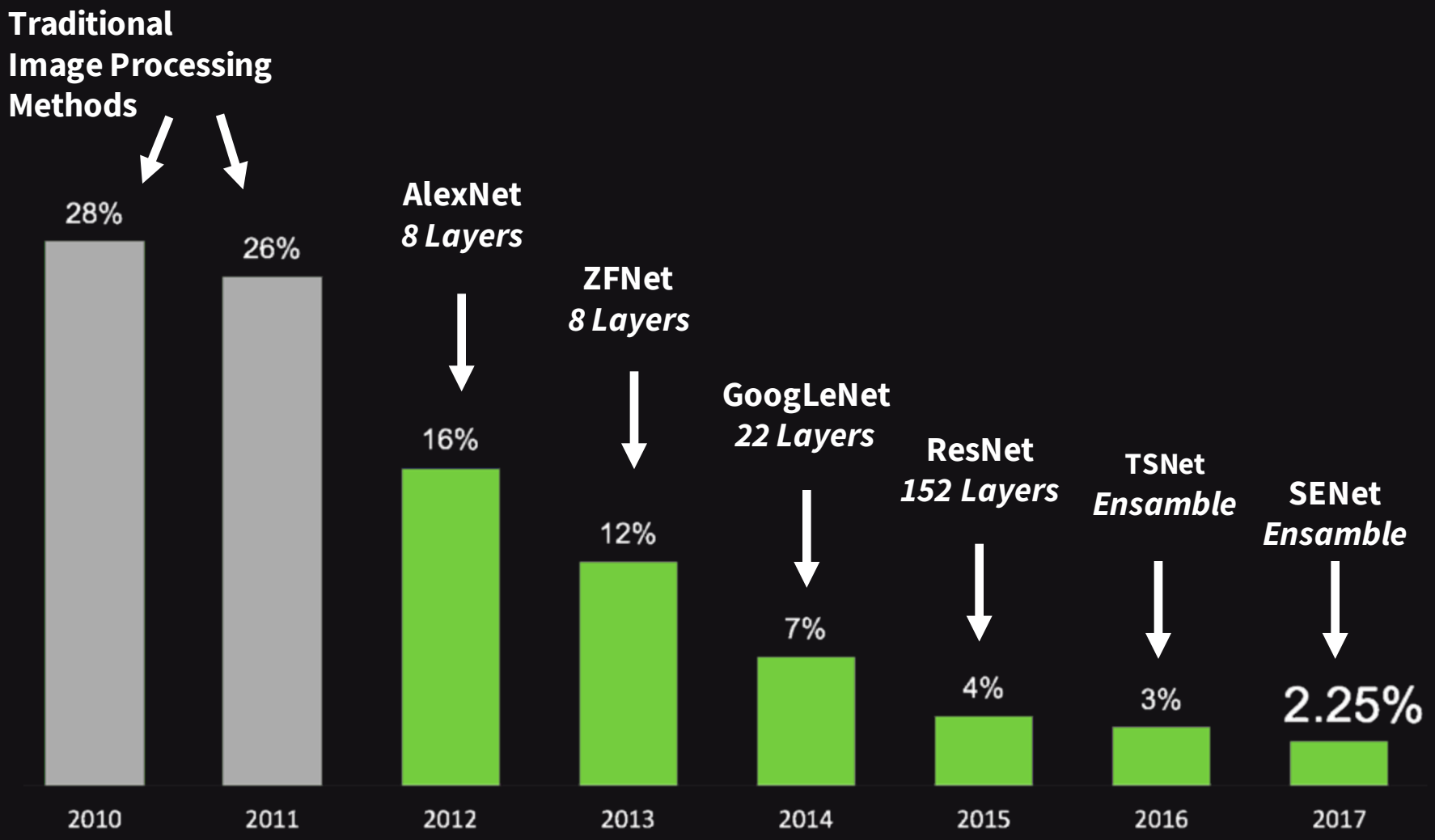


image-net.org/explore
Sausage dog, sausage hound
Informal term
1445 pictures
55.77% Popularity Percentile
Wordnet IDs
invertebrate (766)
homeotherm, homoiotherm,
work animal (4)
darter (0)
survivor (0)
range animal (0)
creepy-crawly (0)
domestic animal, domesticate
domestic cat, house cat,
dog, domestic 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



EXAMPLE: CLASSIFYING PRODUCT IMAGES

Classifying products into 8 categories

- accessories
- jackets
- jeans
- knitwear
- shirts
- shoes
- shorts
- tees

https://github.com/alexcnwy/DeepLearning4ComputerVision

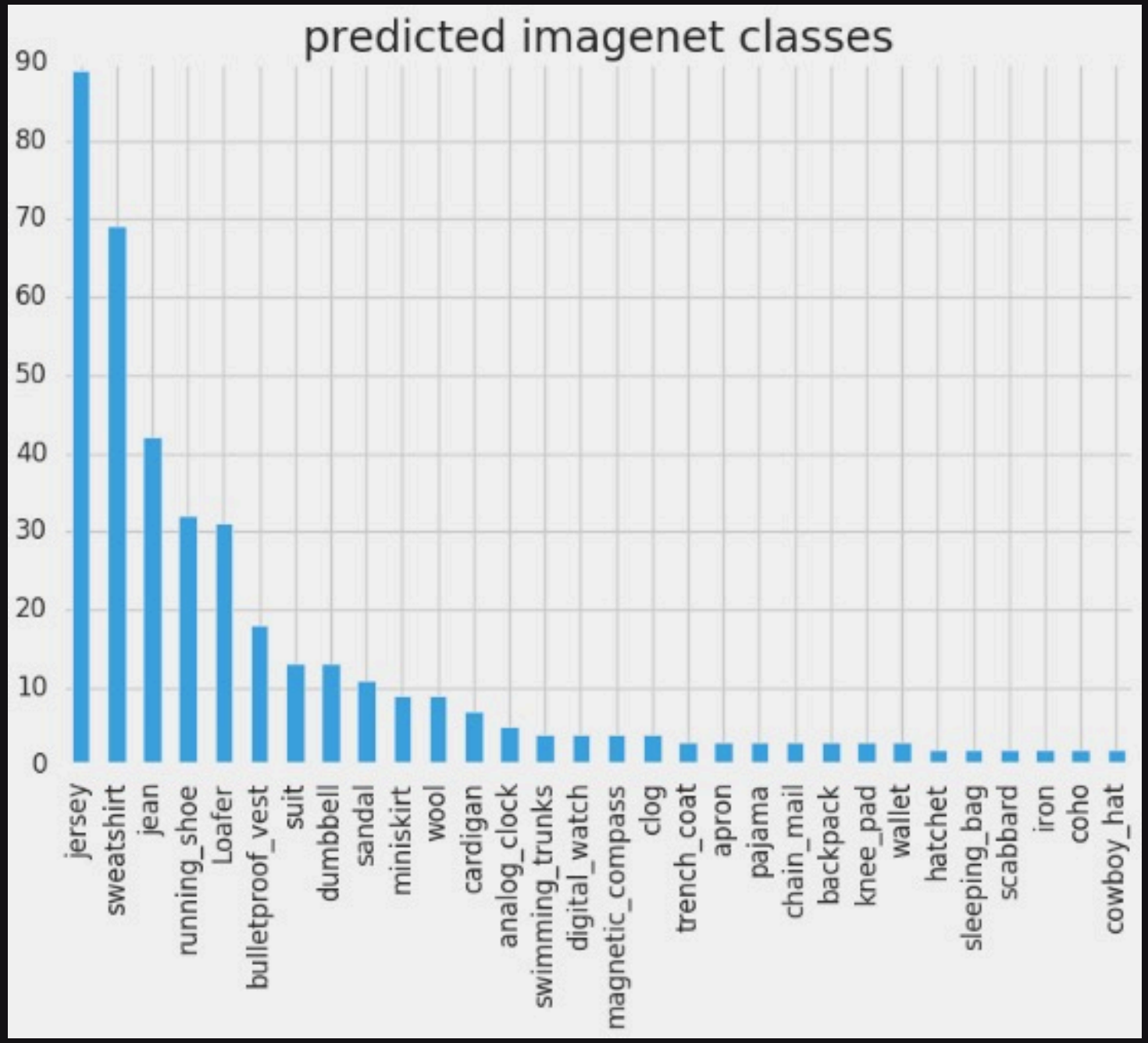
https://github.com/alexcnwy/DeepLearning4ComputerVision

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

BUT
WE WANT TO
PREDICT
SOMETHING NEW!

BEFORE FINE-TUNING

. . .

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)	(None, 1000)	4097000	dropout_4[0][0]

Total params: 138357544

REPLACE FINAL LAYERS OF PRE-TRAINED MODEL

```
# pop layers until just have the bottleneck max pooling 512,7,7 layer
for i in range(0,6):
    model.layers.pop()

model.outputs = [model.layers[-1].output]
model.layers[-1].outbound_nodes = []
```

```
model_top = Sequential()
model_top.add(Flatten(input_shape=train_data.shape[1:]))
model_top.add(Dense(256, activation='relu'))
model_top.add(Dropout(0.5))
model_top.add(Dense(len(np.unique(train_labels)), activation='softmax'))
```

AFTER FINE-TUNING

flatten_4 (Flatten)	(None, 25088)	
dense_8 (Dense)	(None, 256)	
dropout_5 (Dropout)	(None, 256)	
dense_9 (Dense)	(None, 8)	

Fine-tuning a CNN to solve a new problem

- Load pre-trained VGG network
- Remove final dense & softmax layers that predict 1000 ImageNet classes
- Replace with new dense & softmax layer to predict 8 categories

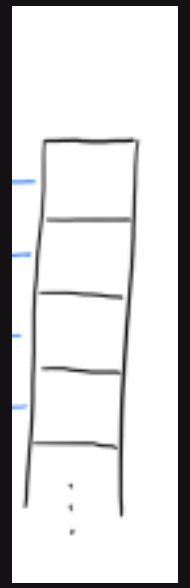
```
Epoch 59/100
3467/3467 [==============================] - 1s - loss: 0.0512 - acc: 0.9787 - val_loss: 0.2634 - val_acc: 0.9503
Epoch 60/100
3467/3467 [==============================] - 1s - loss: 0.0803 - acc: 0.9703 - val_loss: 0.2163 - val_acc: 0.9529
Epoch 61/100
3467/3467 [==============================] - 1s - loss: 0.0496 - acc: 0.9824 - val_loss: 0.2223 - val_acc: 0.9634
```

96.3% accuracy in *under 2 minutes* for classifying products into categories

(WITH ONLY 3467 TRAINING IMAGES!!1!)

f (

) =



"ENCODING"

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

Future Tech / Artificial Intelligence /

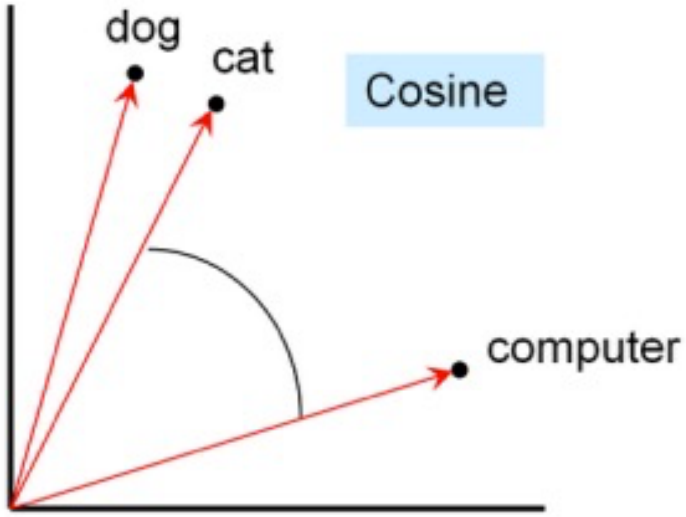
Spree adds AI-powered image search to iOS app

By Andy Walker: Editor on 8 June, 2017 AndyWalkerSA

Cosine Similarity

- Cosine distance: borrowed from information retrieval

$$\text{sim}_{\text{cosine}}(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$



```
def get_most_similar(test_img_path):
    test_img_vec = get_activation(test_img_path)

    # do dot prod
    test_img_vec = test_img_vec.reshape(len(test_img_vec),1)
    a = np.dot(activations, test_img_vec)

    # transform scores
    results = pd.DataFrame(a)
    results['filenames'] = adf['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]

    return matches
```

https://github.com/alexcnwy/DeepLearning4ComputerVision

Input Image

not seen by model

Results

Top 10 most

“visually similar”

Input Image

not seen by model

Results

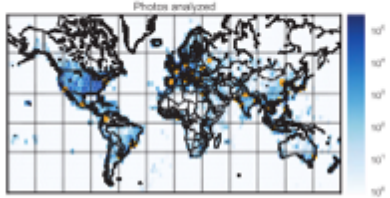
Top 10 most

“visually similar”

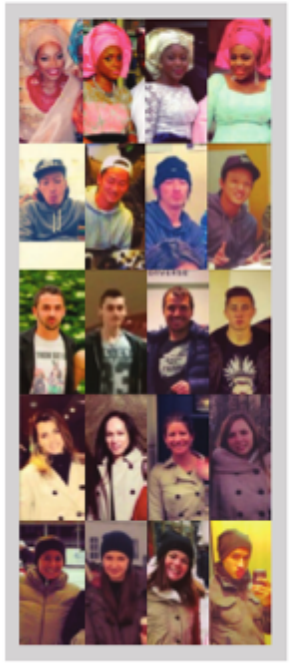
StreetStyle: Exploring world-wide clothing styles from millions of photos

Kevin Matzen * Kavita Bala † Noah Snavely ‡

Cornell University



(a) Massive dataset of people



(b) Global clusters

	No	Yes
Wearing Jacket	18078	7113
Collar Presence	16774	7299
Wearing Scarf	23979	1452
Wearing Necktie	24843	827
Wearing Hat	23279	2255
Wearing Glasses	22058	3401
Multiple Layers	15921	8829

Major Color
Black (6545)
White (4461)
2+ colors (2439)
Blue (2419)
Gray (1345)
Red (1131)
Pink (649)
Green (526)
Yellow (441)
Brown (386)
Purple (170)
Orange (162)
Cyan (33)

Clothing Category
Shirt (4666)
Outerwear (4580)
T-shirt (4580)
Dress (2558)
Tank top (1348)
Suit (1143)
Sweater (874)

Sleeve Length
Long sleeve (13410)
Short sleeve (7145)
No sleeve (3520)

Neckline Shape
Round (9799)
Folded (8119)
V-shape (2017)

Clothing Pattern
Solid (15933)
Graphics (3832)
Striped (1069)
Floral (885)
Plaid (532)
Spotted (241)

Year 2010

NEC-UIUC

Dense grid descriptor: HOG, LBP

Coding: local coordinate, super-vector

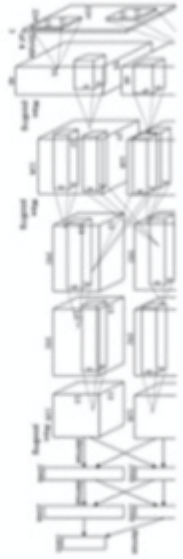
Pooling, SPM

Linear SVM

[Lin CVPR 2011]

Year 2012

SuperVision



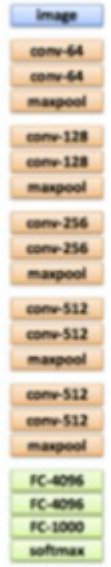
[Krizhevsky NIPS 2012]

Year 2014

GoogLeNet

[Szegedy arxiv 2014]

VGG



[Simonyan arxiv 2014]

Year 2015

MSRA

Computer Science > Machine Learning

Neural Architecture Search with Reinforcement Learning

Barret Zoph, Quoc V. Le

(Submitted on 5 Nov 2016 (v1), last revised 15 Feb 2017 (this version, v2))

Neural networks are powerful and flexible models that work well for many difficult learning tasks in image, speech and natural language understanding. Despite their success, neural networks are still hard to design. In this paper, we use a recurrent network to generate the model descriptions of neural networks and train this RNN with reinforcement learning to maximize the expected accuracy of the generated architectures on a validation set. On the CIFAR-10 dataset, our method, starting from scratch, can design a novel network architecture that rivals the best human-invented architecture in terms of test set accuracy. Our CIFAR-10 model achieves a test error rate of 3.65, which is 0.09 percent better and 1.05x faster than the previous state-of-the-art model that used a similar architectural scheme. On the Penn Treebank dataset, our model can compose a novel recurrent cell that outperforms the widely-used LSTM cell, and other state-of-the-art baselines. Our cell achieves a test set perplexity of 62.4 on the Penn Treebank, which is 3.6 perplexity better than the previous state-of-the-art model. The cell can also be transferred to the character language modeling task on PTB and achieves a state-of-the-art perplexity of 1.214.

https://arxiv.org/abs/1611.01578

WHAT ABOUT VIDEO?

Fig. 9. Performance of different combinations of the 3D CNN architectures. The AUC scores at FPR = 0.1 and 1 percent are multiplied by 10^5 and 10^3, respectively, for better visualization. See the caption of Table 3 and the text for detailed explanations.

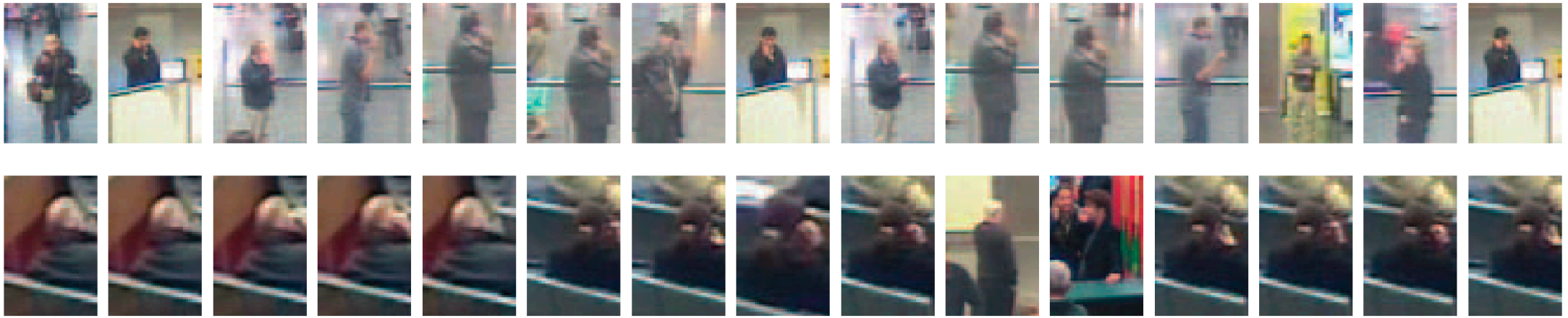
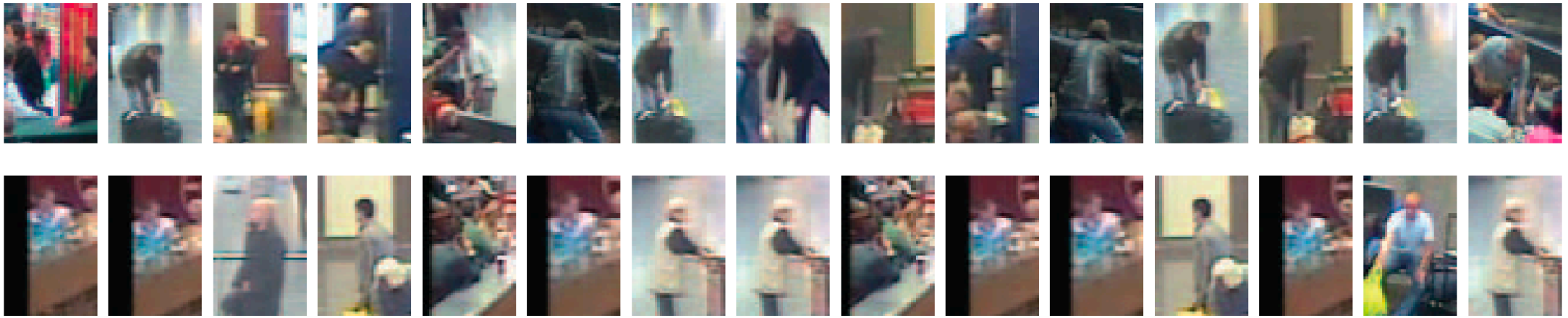


Fig. 10. Sample actions in the CellToEar class. The top row shows actions that are correctly recognized by the combined 3D CNN model, while the bottom row shows those that are misclassified by the model.

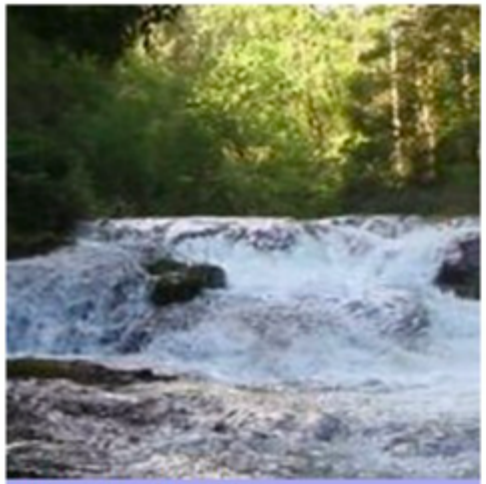


heptathlon
heptathlon
decathlon
hurdles
pentathlon
sprint (running)

bikejoring
mushing
bikejoring
harness racing
skijoring
carting

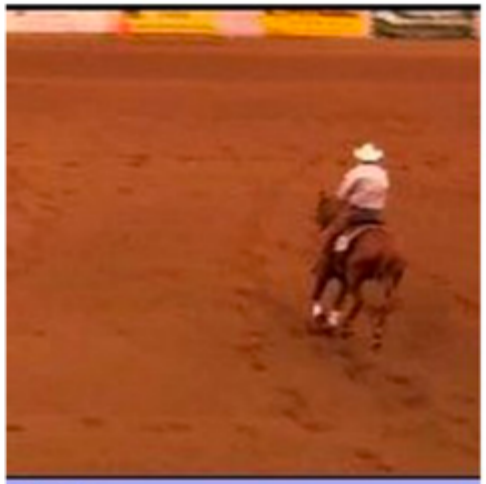
longboarding
longboarding
aggressive inline skating
freestyle scootering
freeboard (skateboard)
sandboarding

ultimate (sport)
ultimate (sport)
hurling
flag football
association football
rugby sevens



whitewater kayaking
whitewater kayaking
rafting
kayaking
canoeing
adventure racing

arena football
indoor american football
arena football
canadian football
american football
women's lacrosse



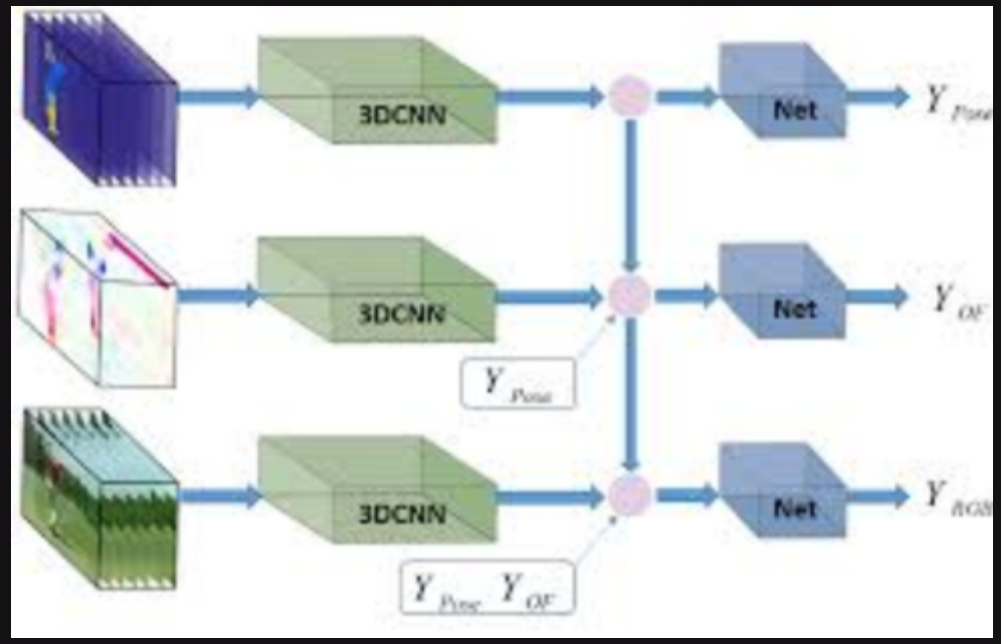
reining
barrel racing
rodeo
reining
cowboy action shooting
bull riding

eight-ball
nine-ball
blackball (pool)
trick shot
eight-ball
straight pool

3D Convolutional Neural Networks for Human Action Recognition

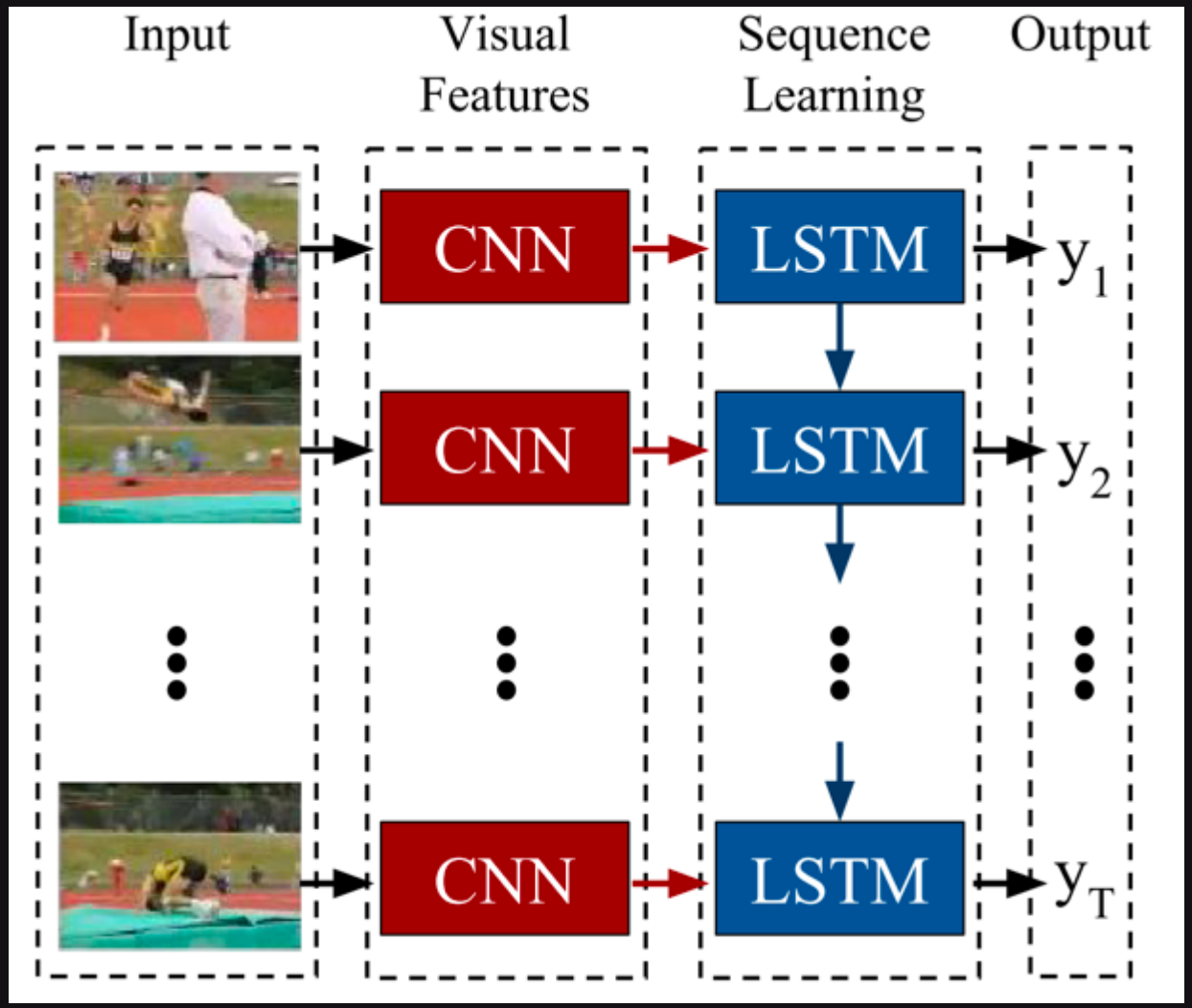
Shuiwang Ji, Wei Xu, Ming Yang, *Member*, *IEEE*, and Kai Yu, *Member*, *IEEE*

"SPATIO-TEMPORAL"



Long-term Recurrent Convolutional Networks for Visual Recognition and Description

Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, Trevor Darrell



"SPATIAL THEN TEMPORAL"

2015

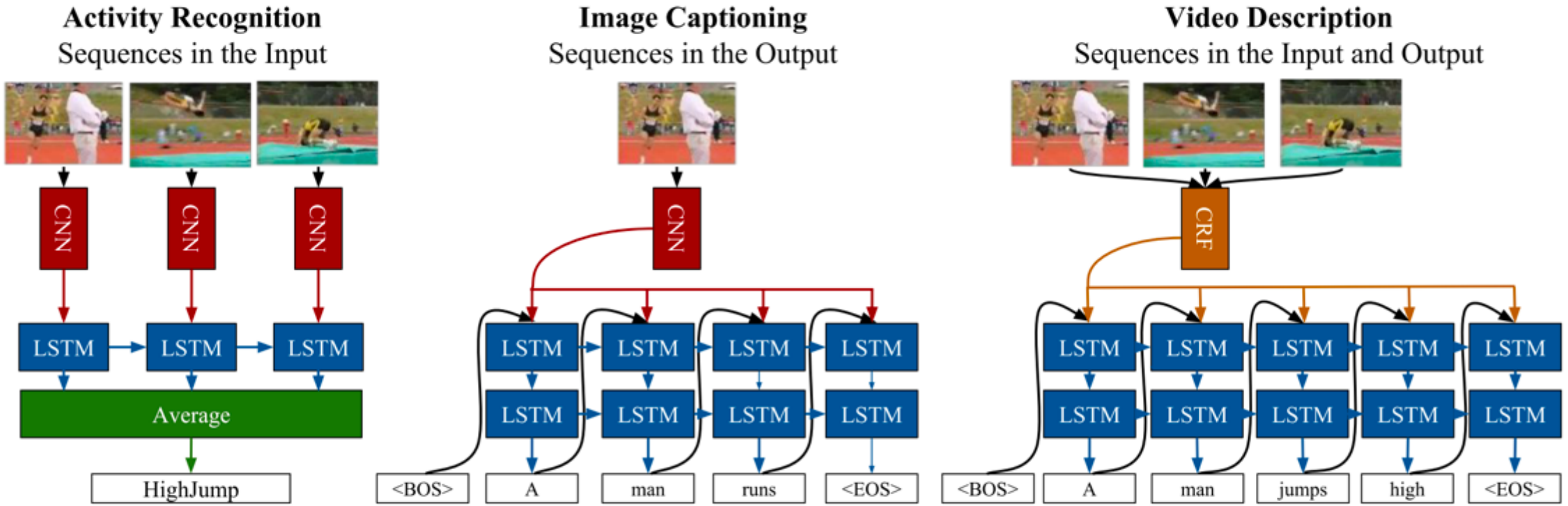
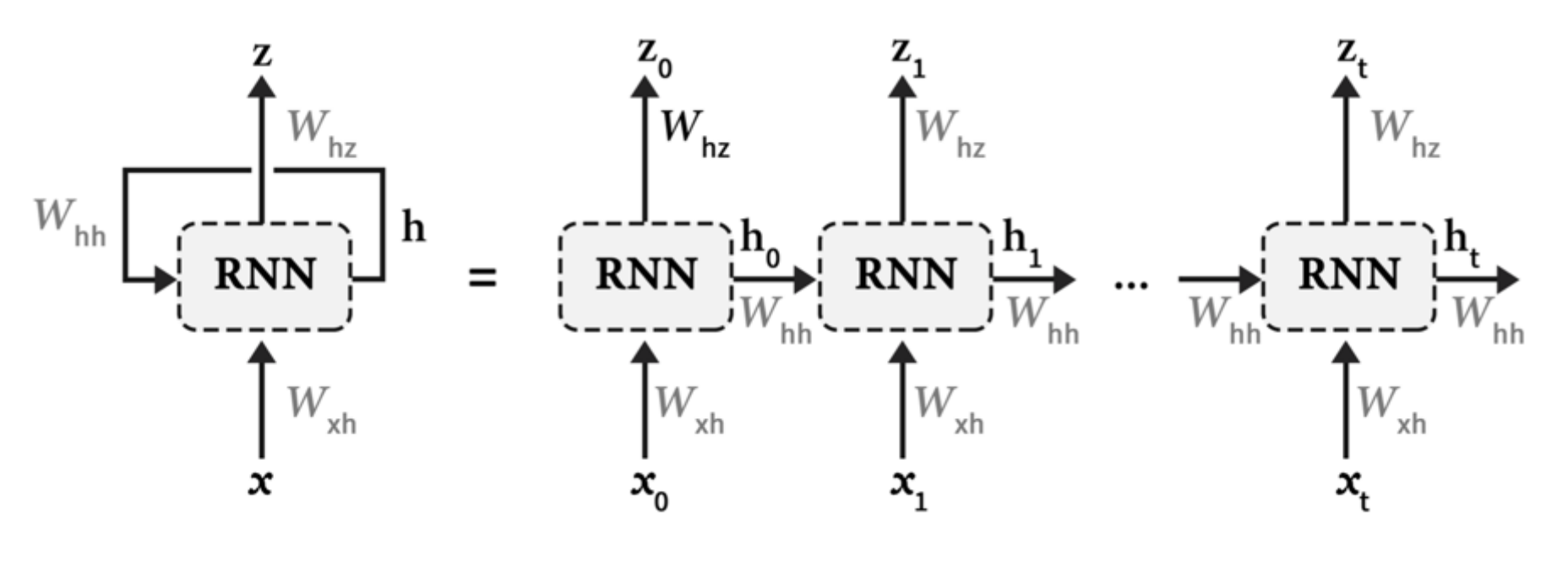


Fig. 3. Task-specific instantiations of our LRCN model for activity recognition, image description, and video description.

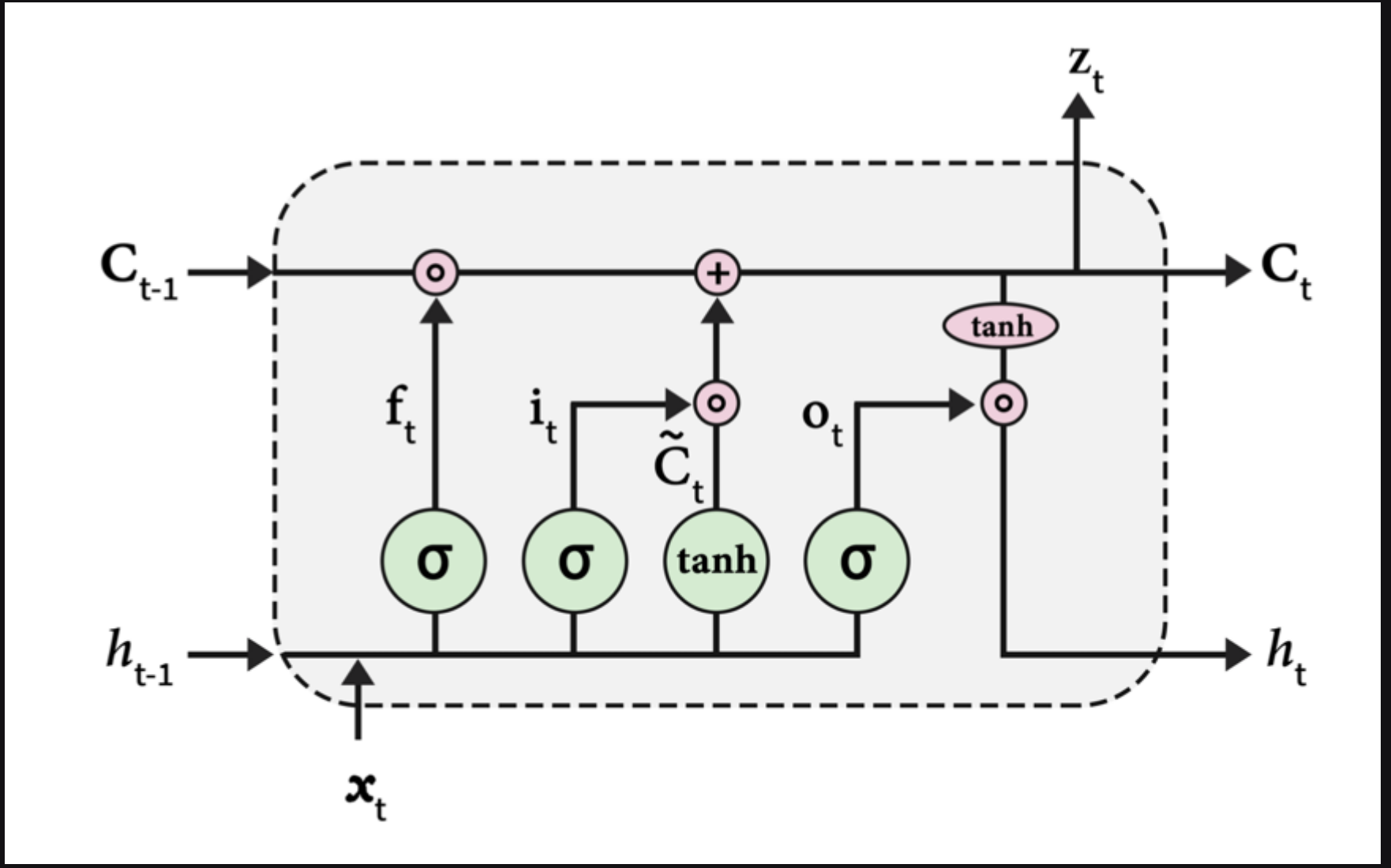
RECURRENT NEURAL NETWORKS (RNNS)

RECURRENT NEURAL NETWORKS

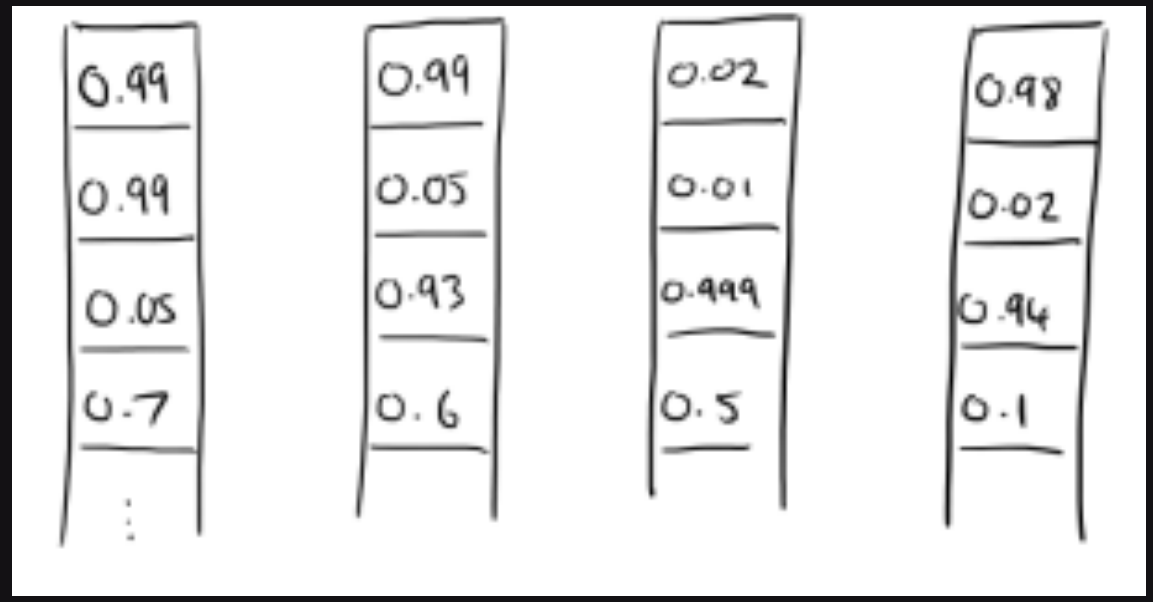


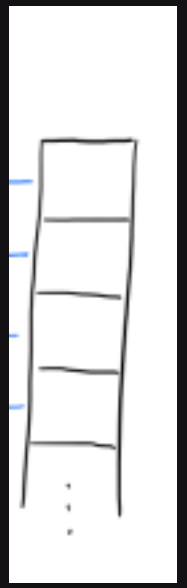
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LONG SHORT TERM MEMORY NETWORKS



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

f(

) = 

"ENCODING"

KERAS
=
CHEAT CODEZ

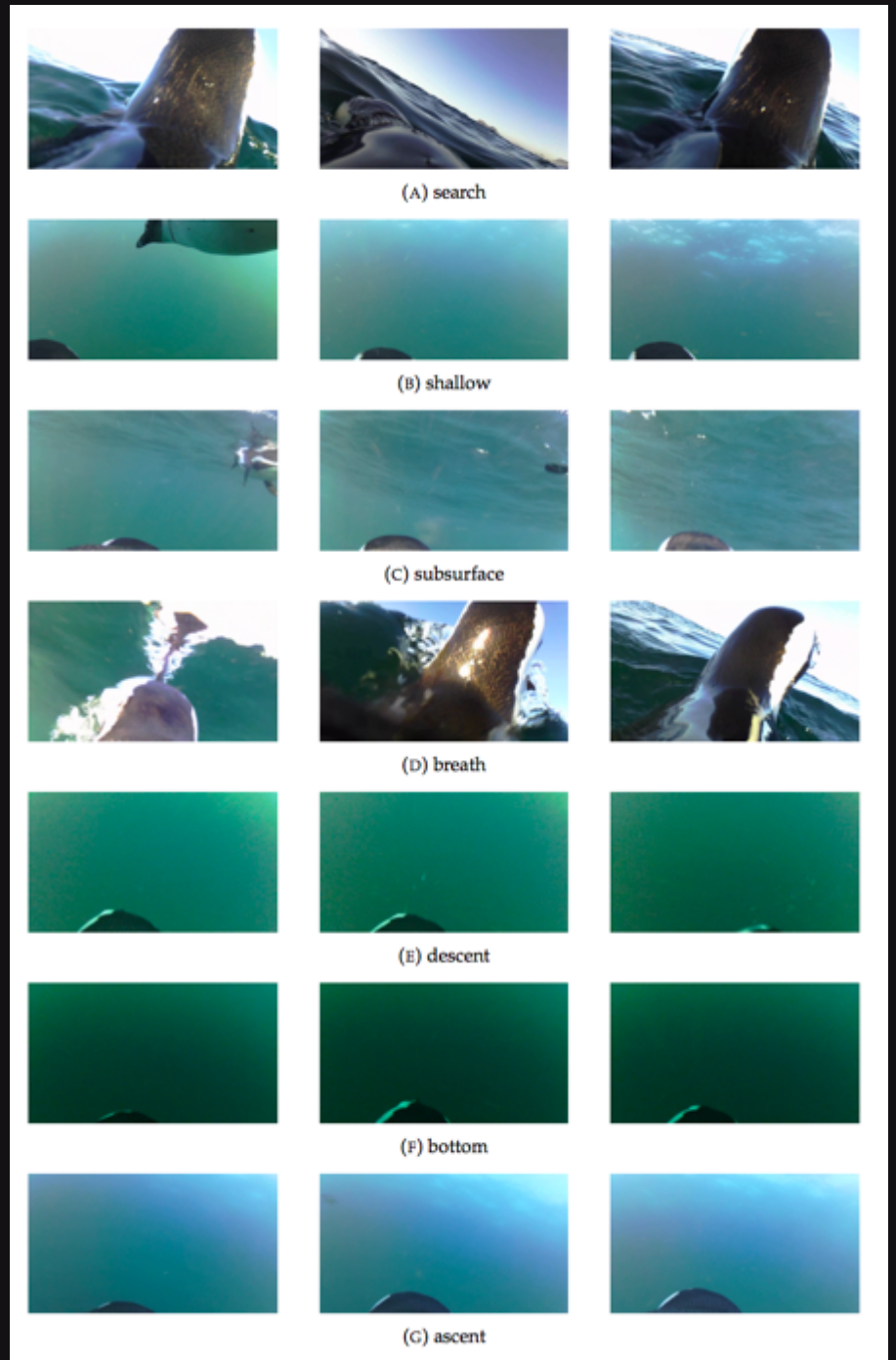
```
'''
from __future__ import print_function

from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding
from keras.layers import LSTM
from keras.datasets import imdb

max_features = 20000
maxlen = 80  # cut texts after this number of words (among top max_features most common words)
batch_size = 32

print('Loading data...')
(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=max_features)
print(len(x_train), 'train sequences')
print(len(x_test), 'test sequences')

print('Pad sequences (samples x time)')
x_train = sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = sequence.pad_sequences(x_test, maxlen=maxlen)
print('x_train shape:', x_train.shape)
print('x_test shape:', x_test.shape)

print('Build model...')
model = Sequential()
model.add(Embedding(max_features, 128))
model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation='sigmoid'))

# try using different optimizers and different optimizer configs
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

print('Train...')
model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=15,
          validation_data=(x_test, y_test))
score, acc = model.evaluate(x_test, y_test,
                            batch_size=batch_size)
print('Test score:', score)
print('Test accuracy:', acc)
```

- https://github.com/keras-team/keras/blob/master/examples/imdb_lstm.py

(A) A feral cat wearing a collar with a modified GoPro Hero 3 video camera and tracking device (Mcgregor et al., 2015)

(B) A tiger shark with a fin-attached Crittercam (Heithaus et al., 2001)

(C) A Crittercam attached using a harness used to investigate the sub-ice foraging behaviour of emperor penguins (Ponganis et al., 2000)



FRAME-LEVEL ACTION RECOGNITION (7 CLASSES)

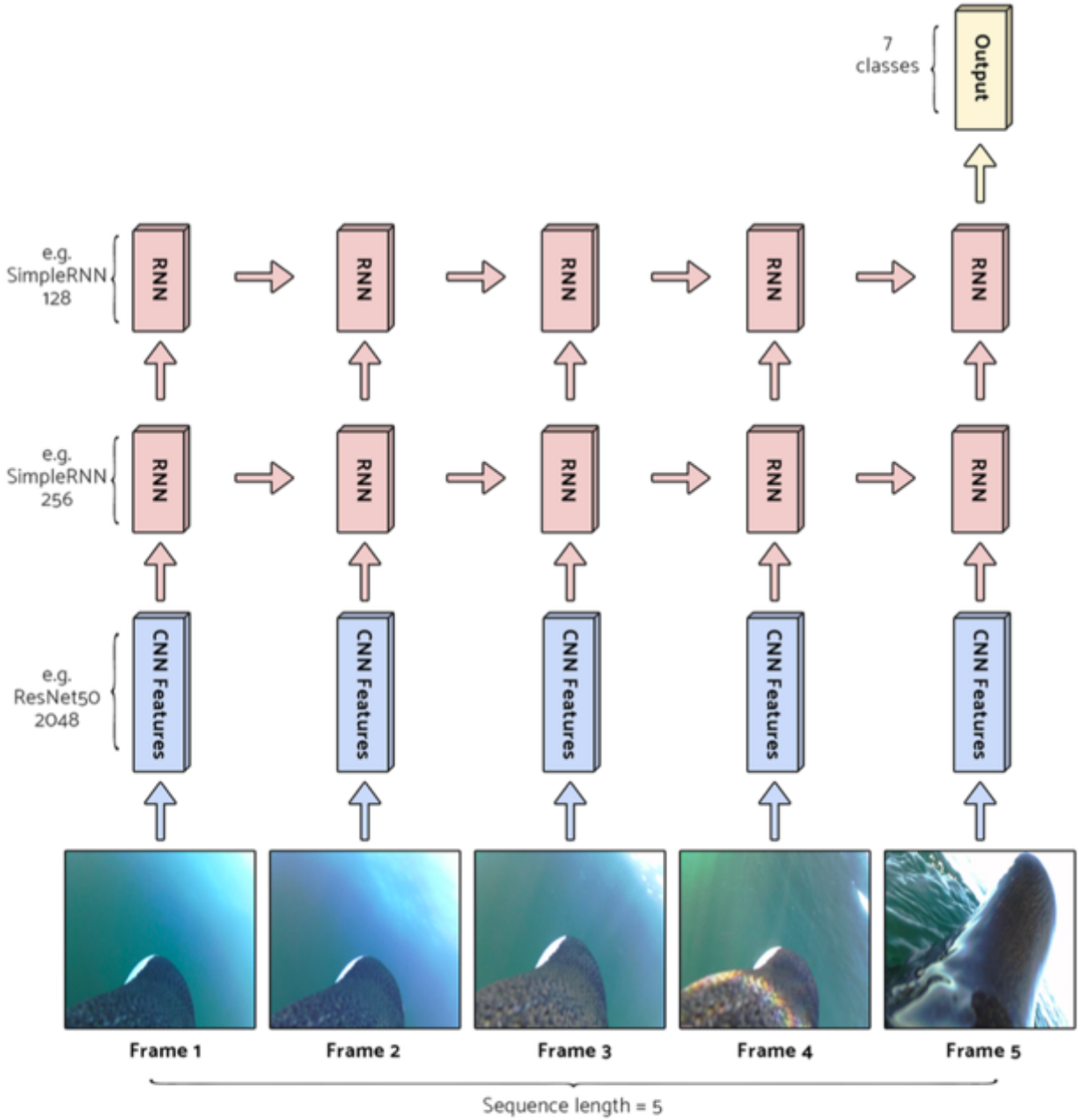


FIGURE 5.3: Modified LRCN Architecture with 2 stacked layers of RNNs applied to penguins dataset with sequence length of 5 using an LSTMs as the RNNs and VGGNet as the CNN

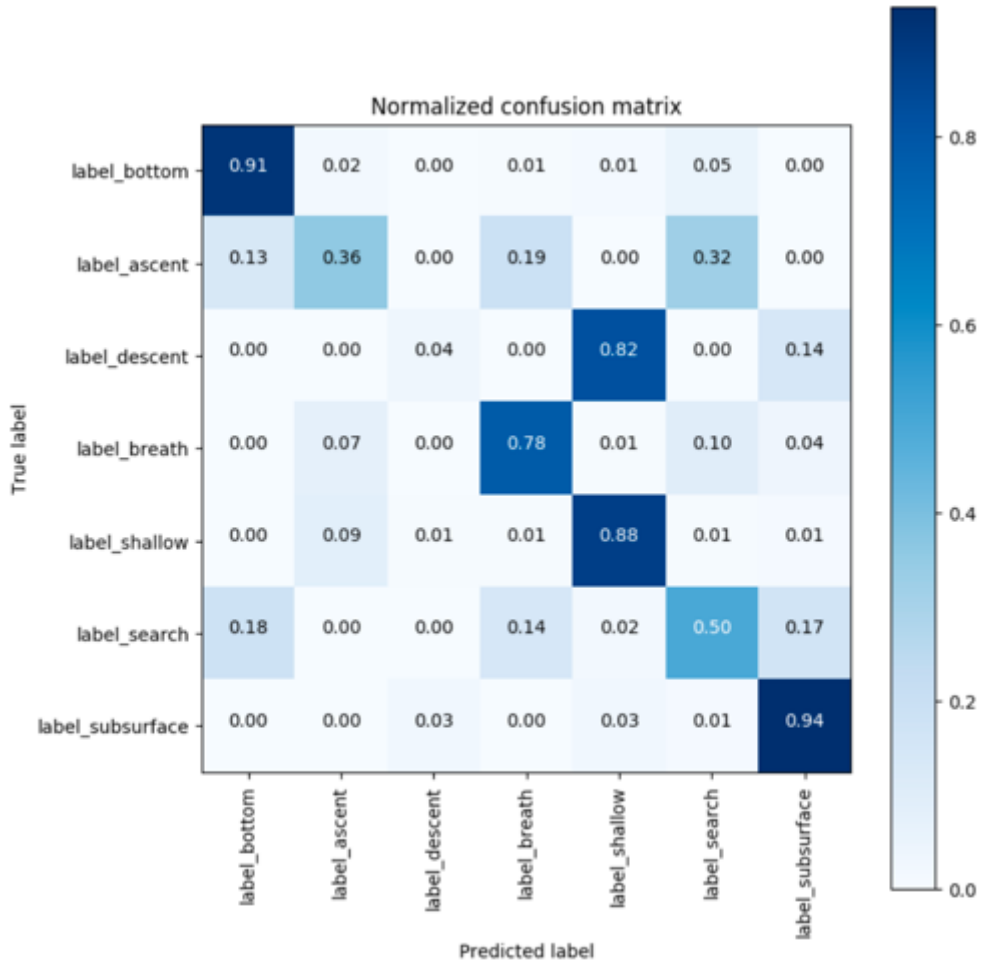
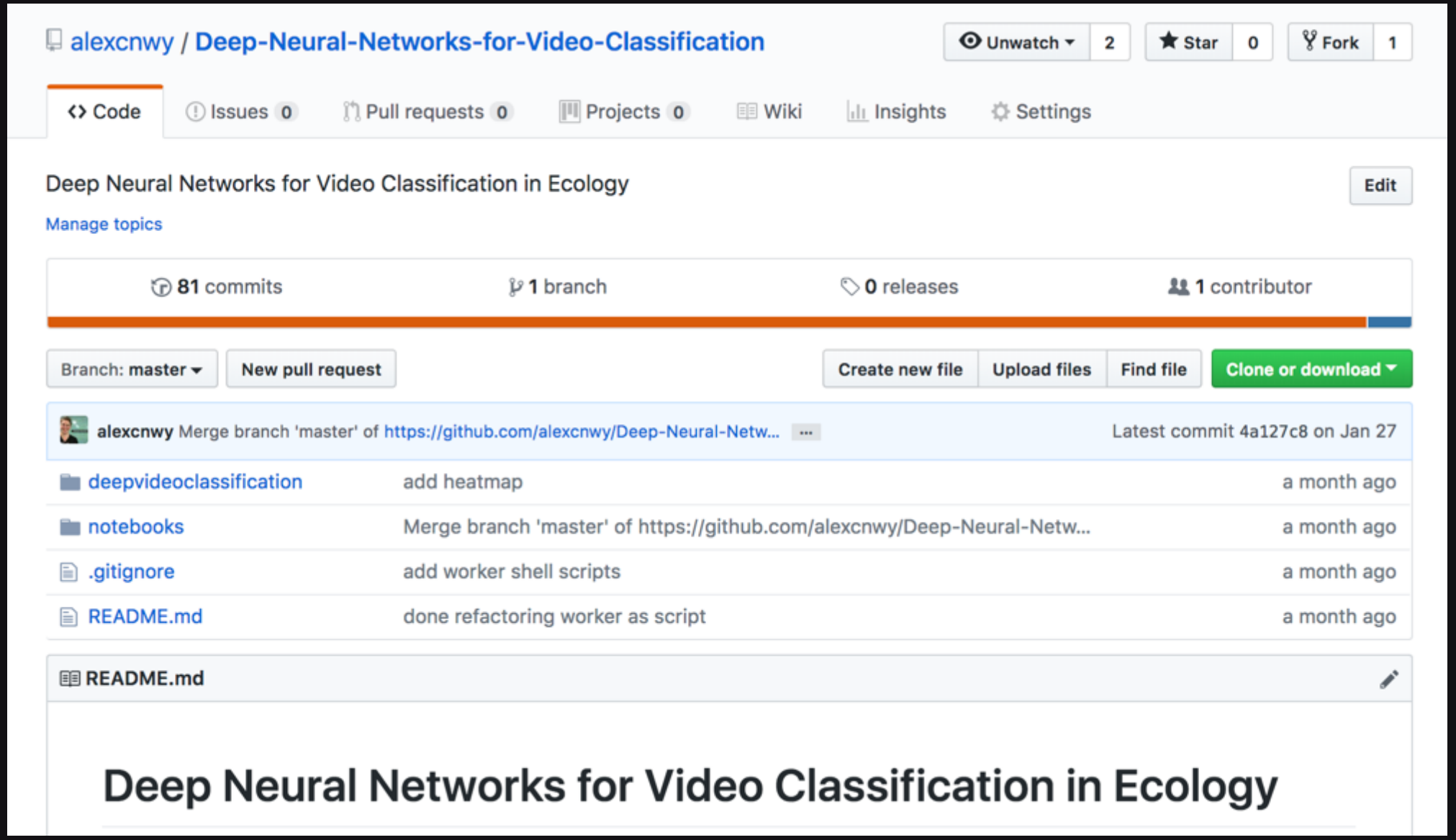


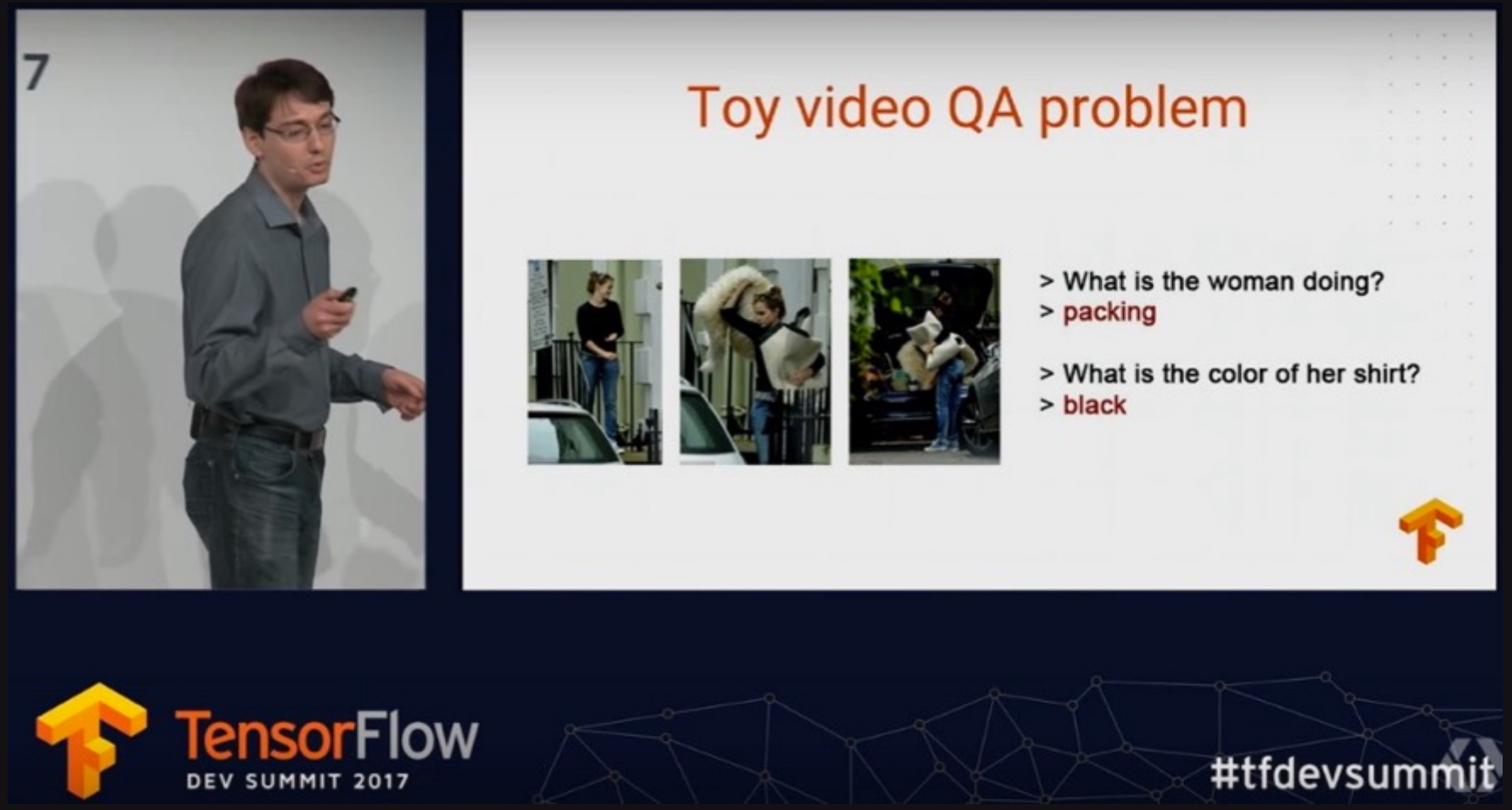
FIGURE 6.1: Normalized confusion matrix for best model on penguins dataset

Accuracy (Validation)	95.87%	95.83%	95.78%	95.77%	95.75%
Accuracy (Train)	98.22%	96.84%	97.50%	96.82%	96.22%
Accuracy (Test)	72.44%	75.67%	72.01%	74.42%	71.82%
Sequence Length	20	20	10	20	10
Architecture	LRCN	LRCN	LRCN	LRCN	LRCN
Train Duration (seconds)	3940	1606	3044	1811	2353
Fit Num Epochs	11	10	13	11	17
Pretrained Model Name	vgg16	vgg16	vgg16	vgg16	vgg16
Num. Features	512	512	512	512	512
Pooling	max	max	max	max	max
Sequence Model	LSTM	SimpleRNN	LSTM	SimpleRNN	SimpleRNN
Sequence Model Layers	2	2	2	2	2
Layer 1 Size	256	256	256	256	256
Layer 2 Size	512	128	512	512	128
Layer 3 Size	256	256	256	256	0
Model Param. Count	4985863	903559	3675143	3214087	247047

TABLE 6.1: Top 5 Most Accurate Models on Validation Set for Penguins Data


alexcnwy / Deep-Neural-Networks-for-Video-Classification
Unwatch 2
Star 0
Fork 1
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings
Deep Neural Networks for Video Classification in Ecology
Edit
Manage topics
81 commits
1 branch
0 releases
1 contributor
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
alexcnwy Merge branch 'master' of https://github.com/alexcnwy/Deep-Neural-Netw...
Latest commit 4a127c8 on Jan 27
deepvideoclassification add heatmap a month ago
notebooks Merge branch 'master' of https://github.com/alexcnwy/Deep-Neural-Netw... a month ago
.gitignore add worker shell scripts a month ago
README.md done refactoring worker as script a month ago
README.md
Deep Neural Networks for Video Classification in Ecology

VIDEO Q&A



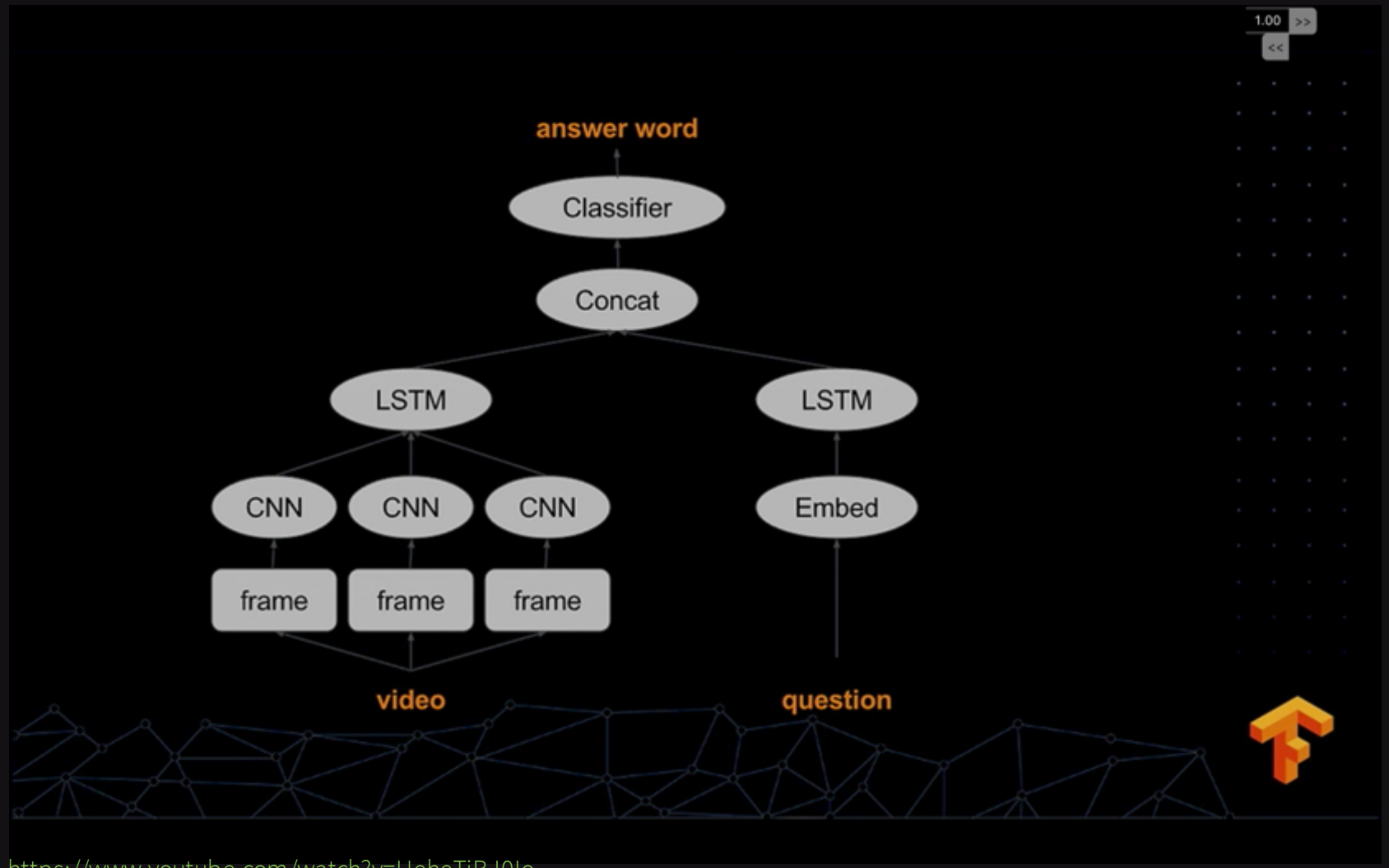
https://www.youtube.com/watch?v=UeheTiBJ0Io



> What is the color of her shirt?
> black

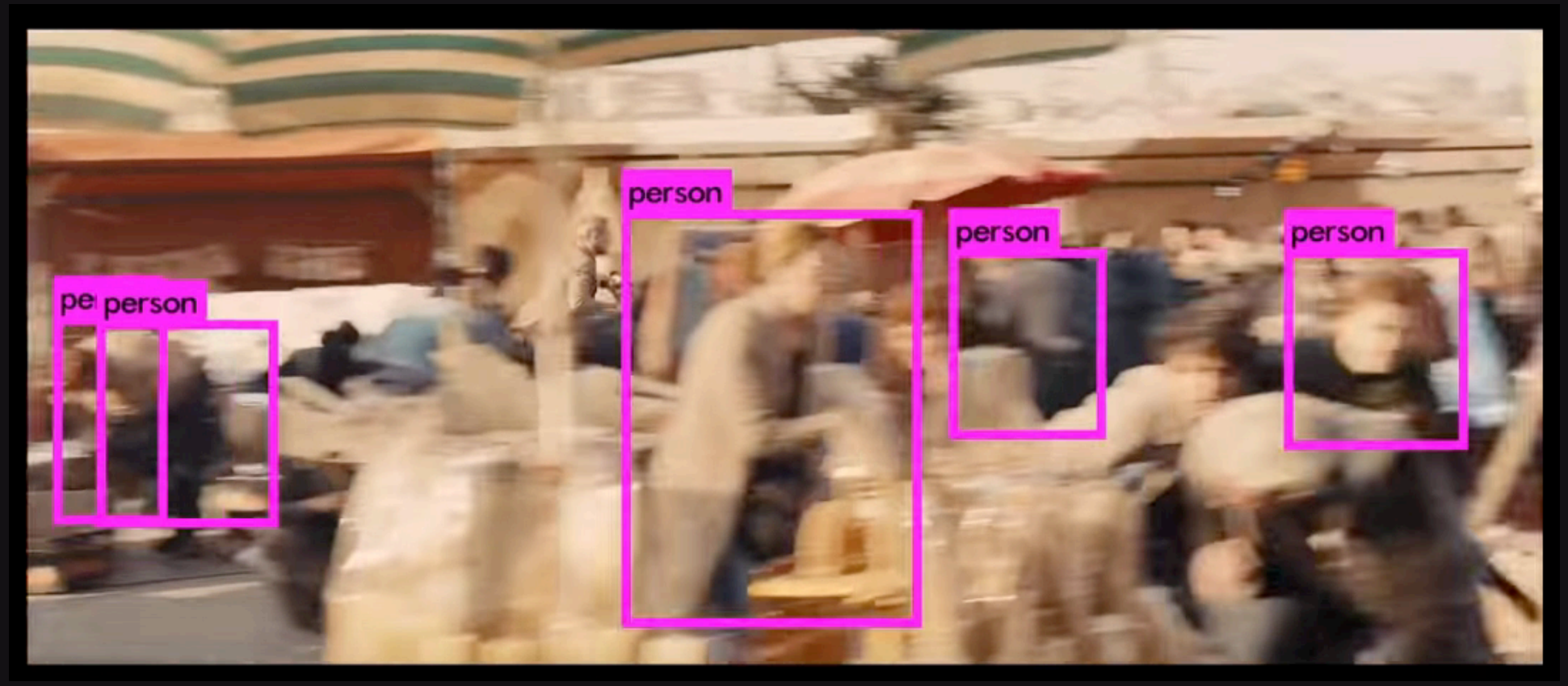
> What is the woman doing?
> packing

VIDEO Q&A

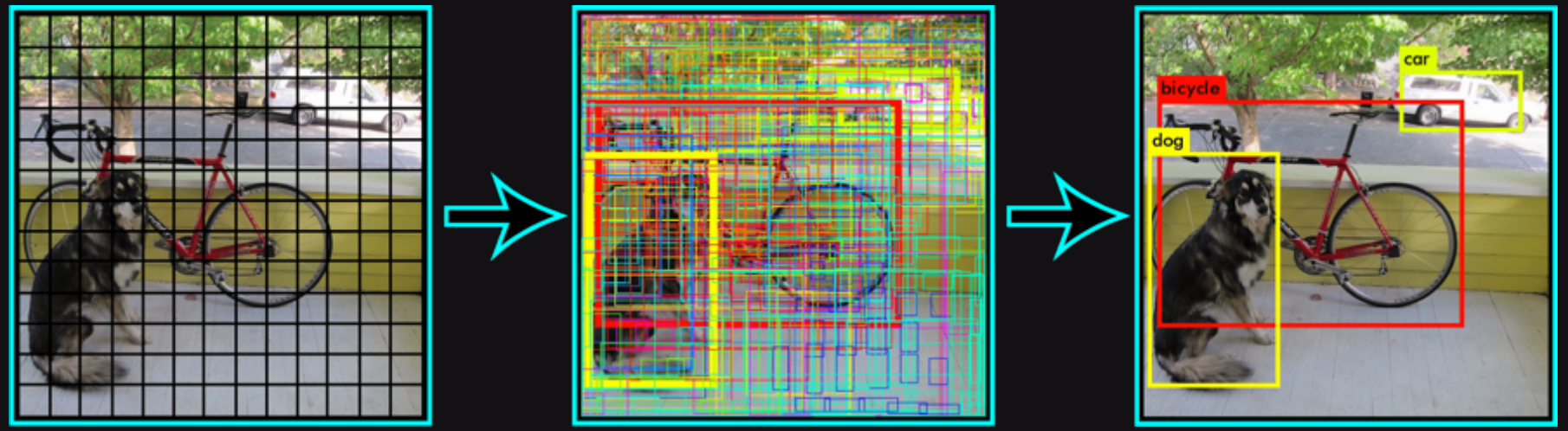


https://www.youtube.com/watch?v=UeheTiBJ0Io

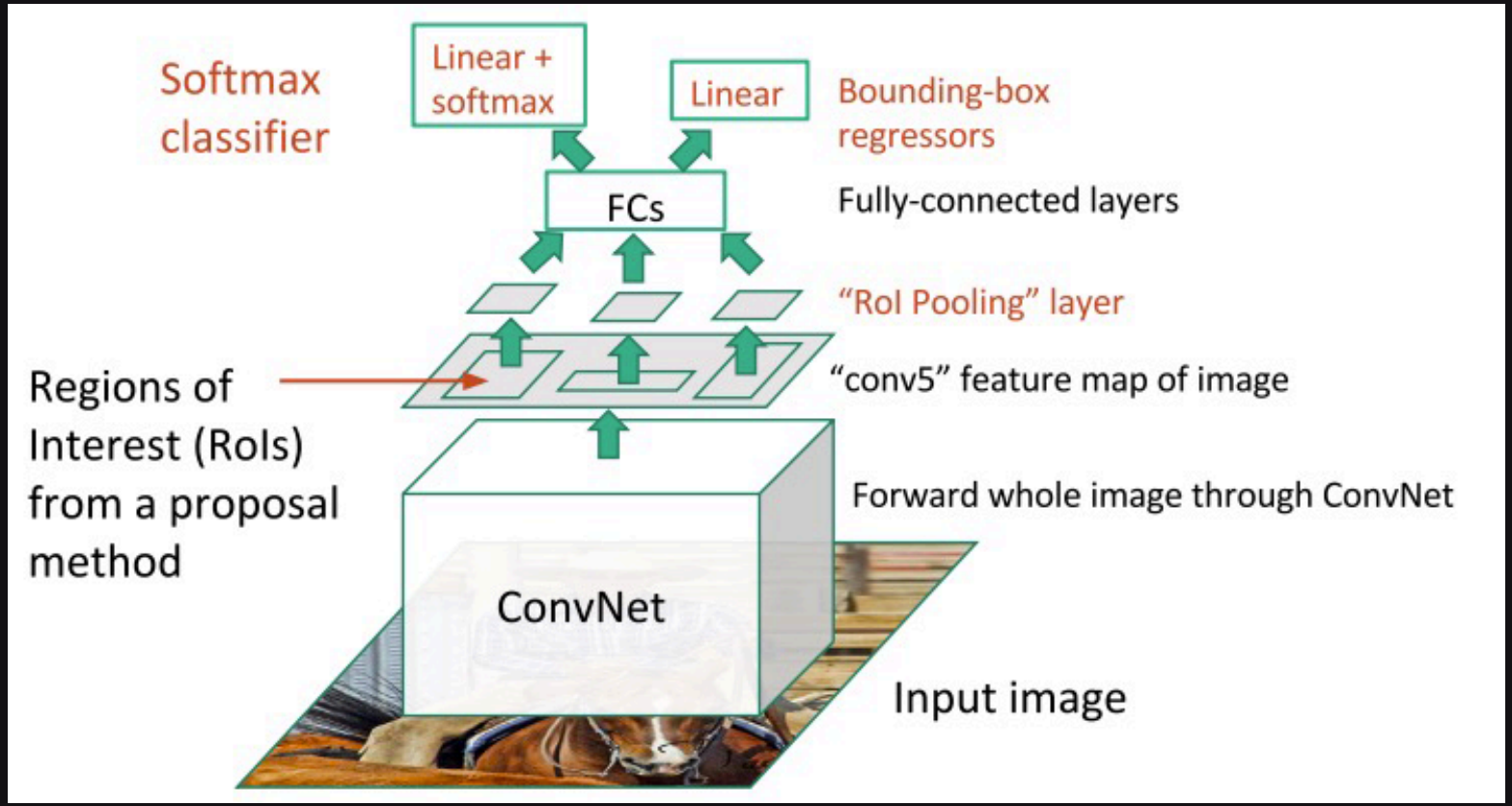
OBJECT DETECTION



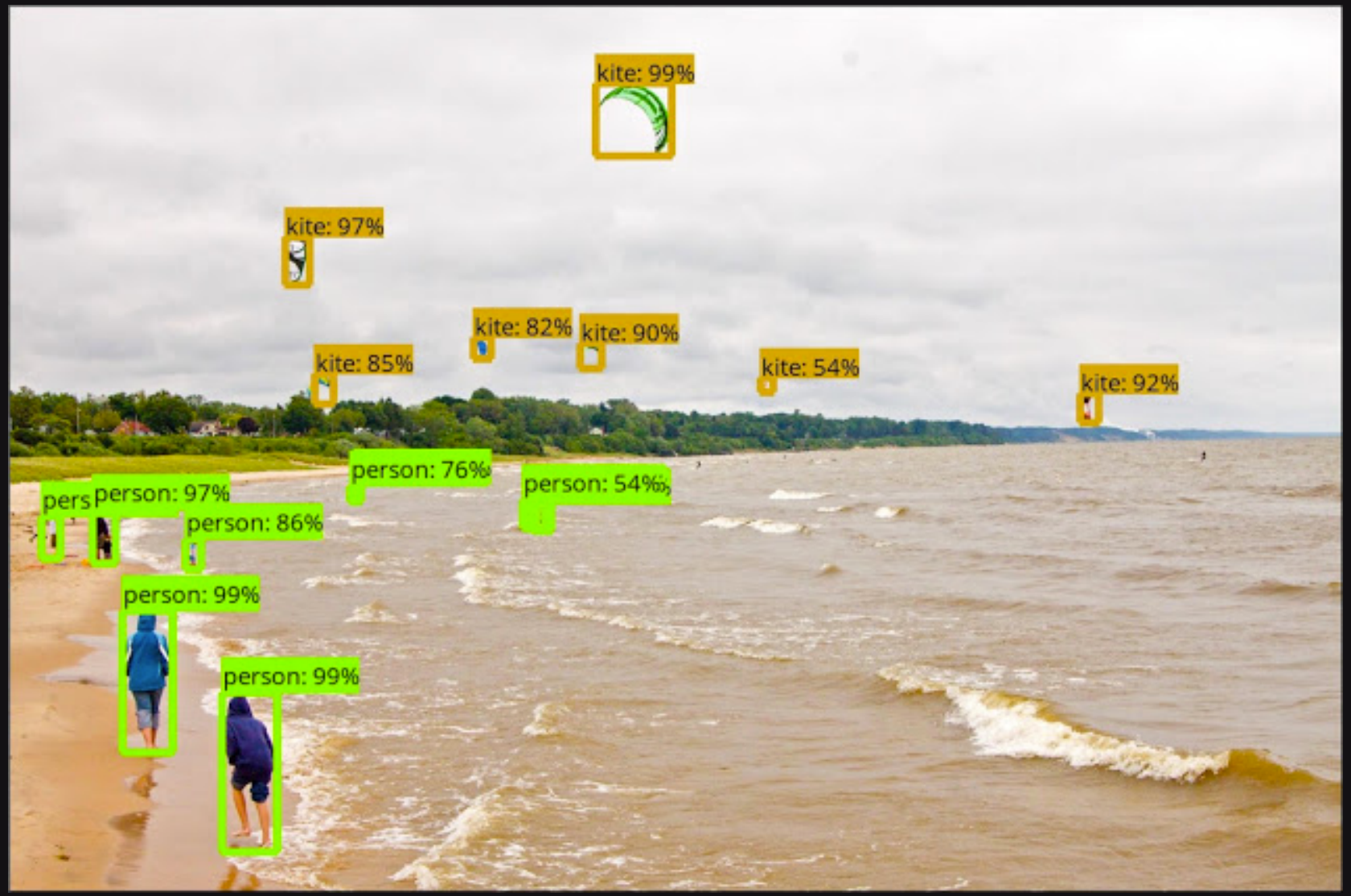
https://www.youtube.com/watch?v=VOC3huqHrss


car
bicycle
dog

OBJECT DETECTION

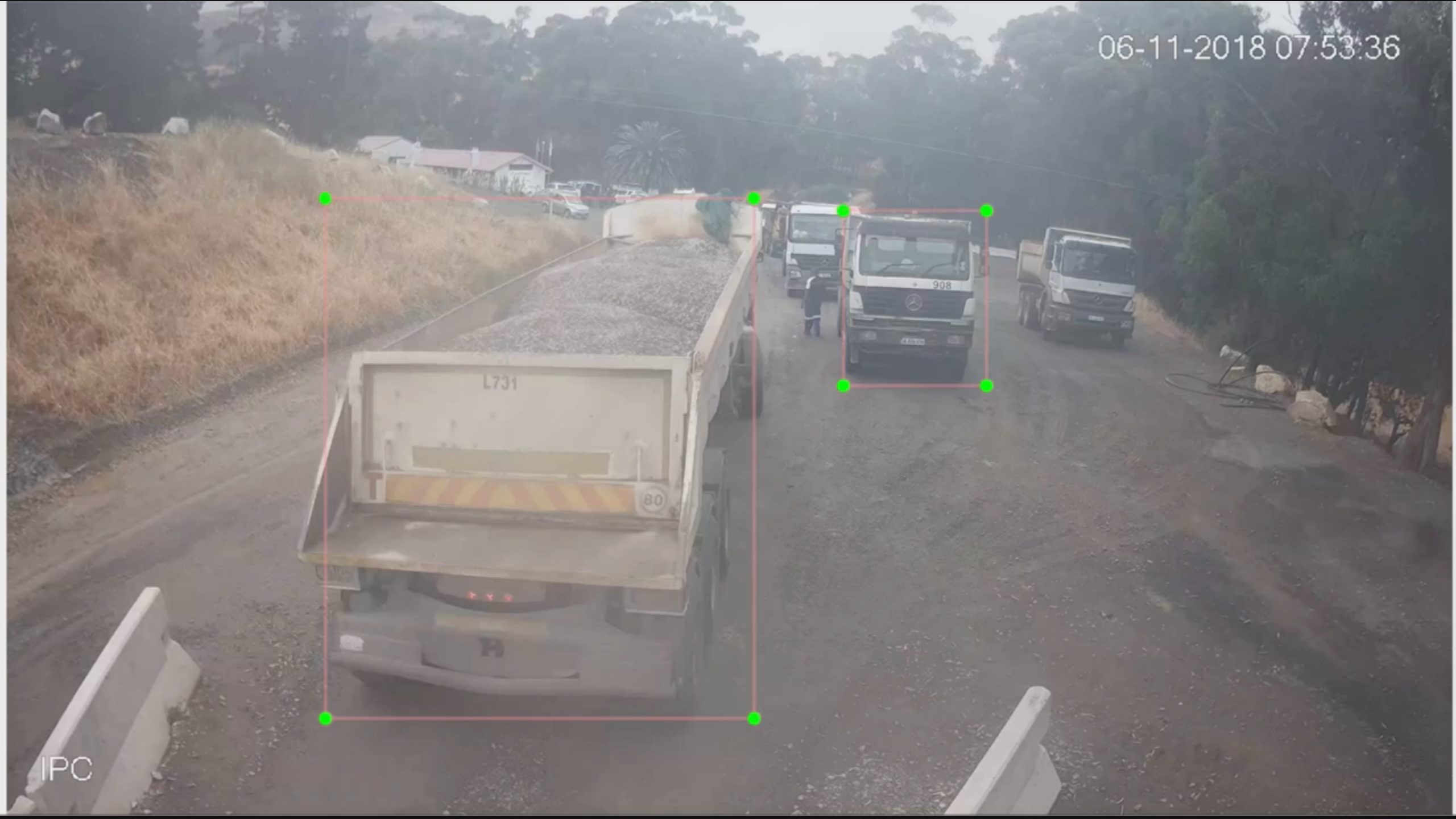


http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf



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

SHOW AND TELL


06-11-2018 07:53:36
L731
908
IPC


06-11-2018 07:53:36
L731
908
OK
IPC


06-11-2018 07:53:36
ANOMALY
OK
ANOMALY
IPC


06-11-2018 07:53:36
L731
908
IPC

Weighbridge

Quarry Inside

Quarry Outside

Bypass Road Inside

Bypass Road Outside

Unresolved Anomalies: 96

Visit Id	Road In	Road Out	Time In	Time Out	In	Out	Anomalies
2201		Bypass Entry		Thu 07 Feb 2019 19:07:05 PM			After-Hours
2186	Bypass Inside		Thu 07 Feb 2019 16:09:59 PM				Road
2182		Bypass Entry		Thu 07 Feb 2019 15:53:38 PM			Road
2181		Bypass Entry		Thu 07 Feb 2019 15:51:27 PM			Road

9:58 4G 68%

81160003290692

[redacted] Anomaly: Truck on Bypass 2018-11-14 08:31:48 [redacted].numberboost.com/visits/546

Tap to load preview

08:34

Text message

SMS

8:36 75%

[redacted].numberboost.com/

Successfully signed in as [redacted]@gmail.com.

Visit 546 (ENTRY)

Time: Wed 14 Nov 2018 08:31:48 AM

Camera: 3

Road: Bypass

Detected Object: truck (Report incorrect type)

Driver: [redacted]

Detected Licence Plate(s): [redacted]

Visit 8836 (ENTRY)

Time: Fri 23 Nov 2018 12:56:53 PM

Camera: 3

Road: Bypass Street

Detected Object: truck (Report Incorrect Type?)

Detected Licence Plate(s): [redacted]

Visit 8836 (ENTRY)

Time: Fri 23 Nov 2018 12:56:53 PM

Camera: 3

Road: Bypass Street

Detected Object: truck (Report Incorrect Type?) TRUCK

Detected Licence Plate(s): [redacted]

Visit 8836 (ENTRY)

Time: Fri 23 Nov 2018 12:56:53 PM

Camera: 3

Road: Bypass Street

Detected Object: truck (Report Incorrect Type?)

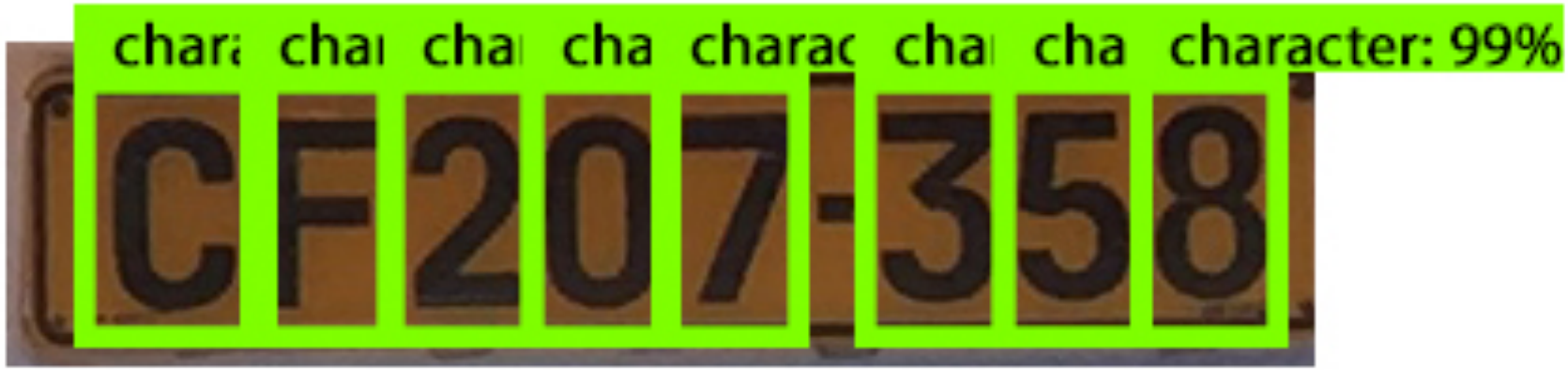
Detected Licence Plate(s): [redacted]

- ✓ TRUCK
- WATER TRUCK
- CAR
- TOP-LOADER
- BULLDOZER
- DIGGER LOADER
- ADD NEW OBJECT

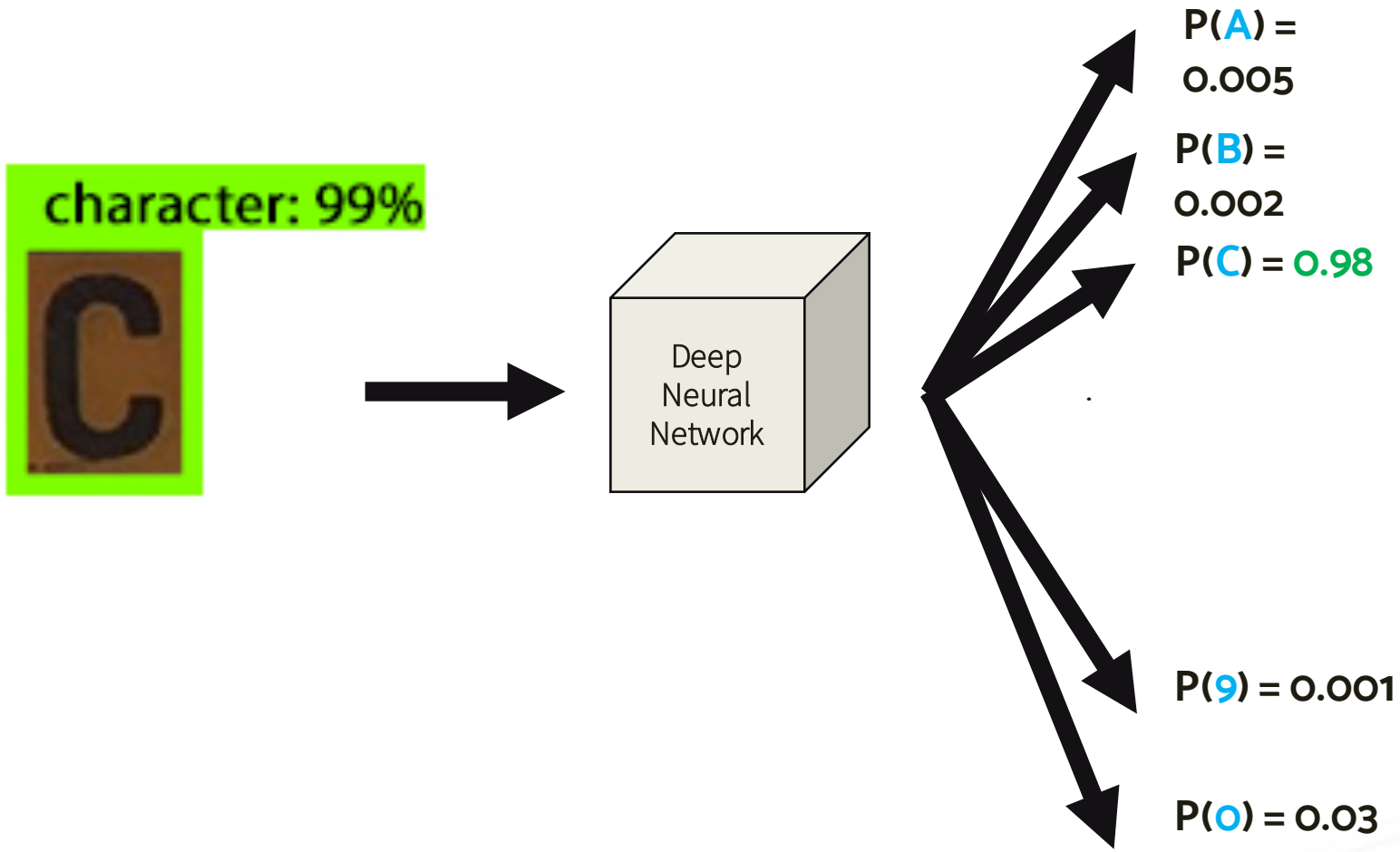

Über Flavour
TEASE
CA 671 934
CF207-358


Über Flavour
TEASE
CA 671 934
CF207-358


CF207 358


chara cha cha cha charac cha cha character: 99%
CF207-358


character: 99%
CF207-358


character: 99%
C
Deep
Neural
Network
P(A) =
0.005
P(B) =
0.002
P(C) = 0.98
P(9) = 0.001
P(O) = 0.03


Über Flavour
TEASE
CA 671 934
CA671934
CF207-358
CF207358

EDGE COMPUTING + FEDERATED ML

TAXI VID DATA:
1 WEEK, 1 CAM = ~25GB

SIZE DOES MATTER

(w,h,3,t)

e.g.
500x500x3x4x60
= 180 million

https://www.reddit.com/r/southafrica/comments/asl4n5/when_a_little_is_just_not_enough/


2018-Nov-02 04:09:37.473 PM (SAST)
No taxi
5
4
3
2
1
15
SW-2960 - Port 1


CO OR!

OFF-THE-SHELF

ML MODELS

DO NOT WORK

IN

AFRICA

Why is ML Succeeding?

1. ~~Algorithms~~
2. ~~Compute~~
3. Data

tensorflow

http://www.tensorflow.org

github-admin@tensorflow.org

```
git clone https://github.com/tensorflow.git
```


https://en.wikipedia.org/wiki/Mechanical_Turk
WIKIPEDIA
The Free Encyclopedia
Main page
Contents
Featured content
Current events
Article
Talk
Mechanical Turk
From Wikipedia, the free encyclopedia
(Redirected from Mechanical turk)
Mechanical Turk may refer to:
The Turk, an 18th-century fake chess-playing machine
Amazon Mechanical Turk, an online crowdsourcing marketplace platform

The Turk

From Wikipedia, the free encyclopedia

This article is about the chess-playing automaton. For other uses, see Turk (disambiguation).

The Turk, also known as the **Mechanical Turk** or **Automaton Chess Player** (German: *Schachtürke*, "chess Turk"; Hungarian: *A Török*), was a fake chess-playing machine constructed in the late 18th century. From 1770 until its destruction by fire in 1854 it was exhibited by various owners as an automaton, though it was eventually revealed to be an elaborate hoax.[1] Constructed and unveiled in 1770 by Wolfgang von Kempelen (Hungarian: Kempelen Farkas; 1734–1804) to impress the Empress Maria Theresa of Austria, the mechanism appeared to be able to play a strong game of chess against a human opponent, as well as perform the knight's tour, a puzzle that requires the player to move a knight to occupy every square of a chessboard exactly once.

The Turk was in fact a mechanical illusion that allowed a human chess master hiding inside to operate the machine. With a skilled operator, the Turk won most of the games played during its demonstrations around Europe and the Americas for nearly 84 years, playing and defeating many challengers including statesmen such as Napoleon Bonaparte and Benjamin Franklin. The device was later purchased in 1804 and exhibited by Johann Nepomuk Mälzel. The chess masters who secretly operated it included Johann Allgaier, Boncourt, Aaron Alexandre, William Lewis, Jacques Mouret, and William Schlumberger, but the operators within the mechanism during Kempelen's original tour remain a mystery.

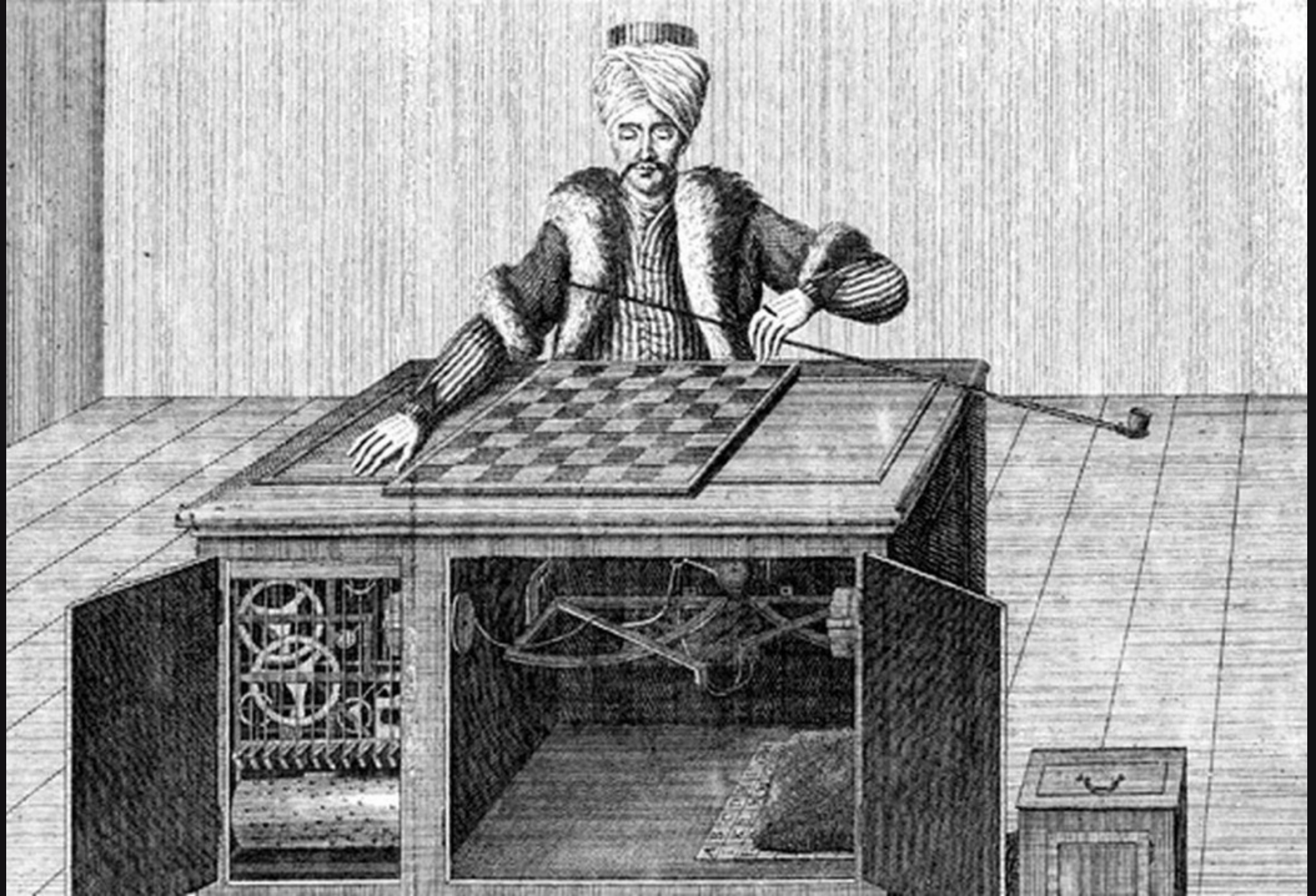
The Turk

From Wikipedia, the free encyclopedia

This article is about the chess-playing automaton. For other uses, see Turk (disambiguation).

The Turk, also known as the **Mechanical Turk** or **Automaton Chess Player** (German: *Schachtürke*, "chess Turk"; Hungarian: *A Török*), was a fake chess-playing machine constructed in the late 18th century. From 1770 until its destruction by fire in 1854 it was exhibited by various owners as an automaton, though it was eventually revealed to be an elaborate hoax.[1] Constructed and unveiled in 1770 by Wolfgang von Kempelen (Hungarian: Kempelen Farkas; 1734–1804) to impress the Empress Maria Theresa of Austria, the mechanism appeared to be able to play a strong game of chess against a human opponent, as well as perform the knight's tour, a puzzle that requires the player to move a knight to occupy every square of a chessboard exactly once.

The Turk was in fact a mechanical illusion that allowed a human chess master hiding inside to operate the machine. With a skilled operator, the Turk won most of the games played during its demonstrations around Europe and the Americas for nearly 84 years, playing and defeating many challengers including statesmen such as Napoleon Bonaparte and Benjamin Franklin. The device was later purchased in 1804 and exhibited by Johann Nepomuk Mälzel. The chess masters who secretly operated it included Johann Allgaier, Boncourt, Aaron Alexandre, William Lewis, Jacques Mouret, and William Schlumberger, but the operators within the mechanism during Kempelen's original tour remain a mystery.



Amazon Mechanical Turk

From Wikipedia, the free encyclopedia

This article's **factual accuracy may be compromised due to out-of-date information**. Please update this article to reflect recent events or newly available information. *(July 2015)*

Amazon Mechanical Turk (**MTurk**) is a crowdsourcing Internet marketplace enabling individuals and businesses (known as Requesters) to coordinate the use of human intelligence to perform tasks that computers are currently unable to do. It is one of the sites of Amazon Web Services, and is owned by Amazon.[2] Employers are able to post jobs known as Human Intelligence Tasks (HITs), such as choosing the best among several photographs of a storefront, writing product descriptions, or identifying performers on music CDs. Workers, colloquially known as *Turkers*, can then browse among existing jobs and complete them in exchange for a monetary payment set by the employer. To place jobs, the requesting programs use an open application programming interface (API), or the more limited MTurk Requester site.[3] To submit a request for tasks to be completed through the Amazon Mechanical Turk web site, a requester must provide a billing address in one of around 30 approved countries.[4]

Amazon Mechanical Turk

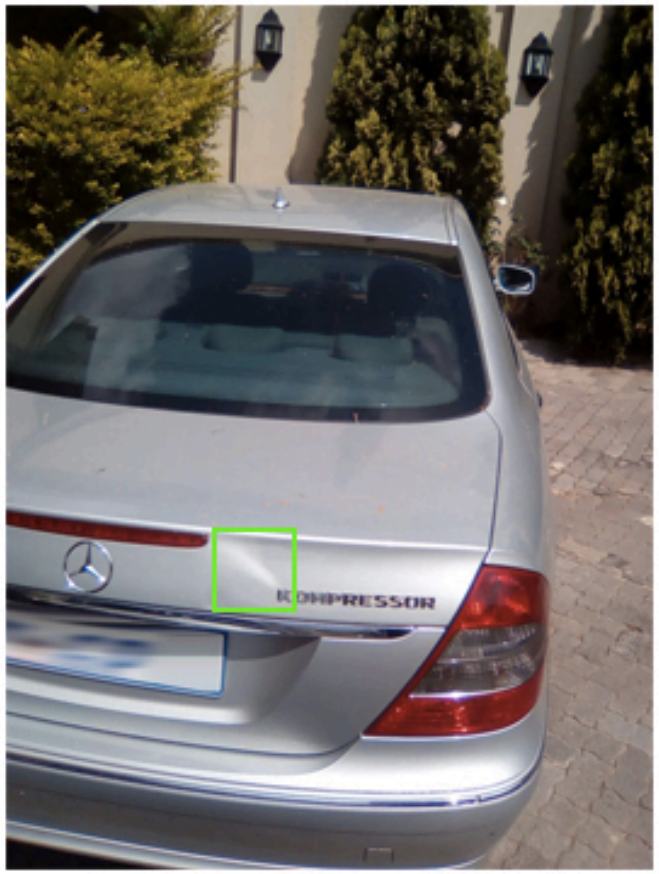
Website	www.mturk.com
Alexa rank	5,330 (September 2016)[1]
Current status	Live

Amazon Mechanical Turk

https://www.mturk.com/mturk/accept?hitId=3 &prevHitSubmitted=f...

Draw a box around any dents or scratches in the photo

Use your mouse to draw a rectangle around any damage on the vehicle in the photo below.



Reset Submit

IMAGENET

14 million images
Labelled by 50'000 Mechanical Turkers

28%

UNEMPLOYMENT!

Translation of *-sebenza* in English:

-sebenza

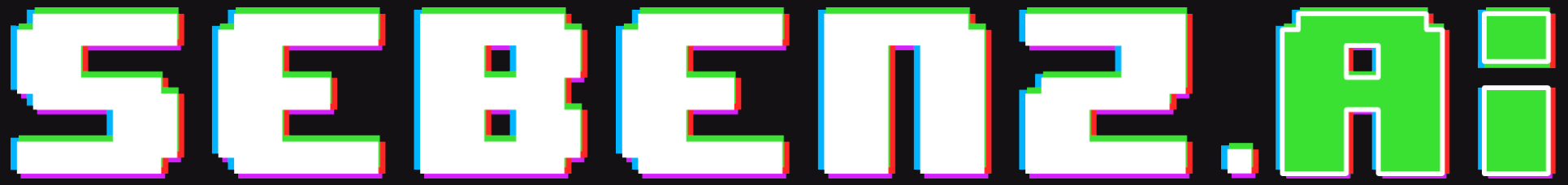
work

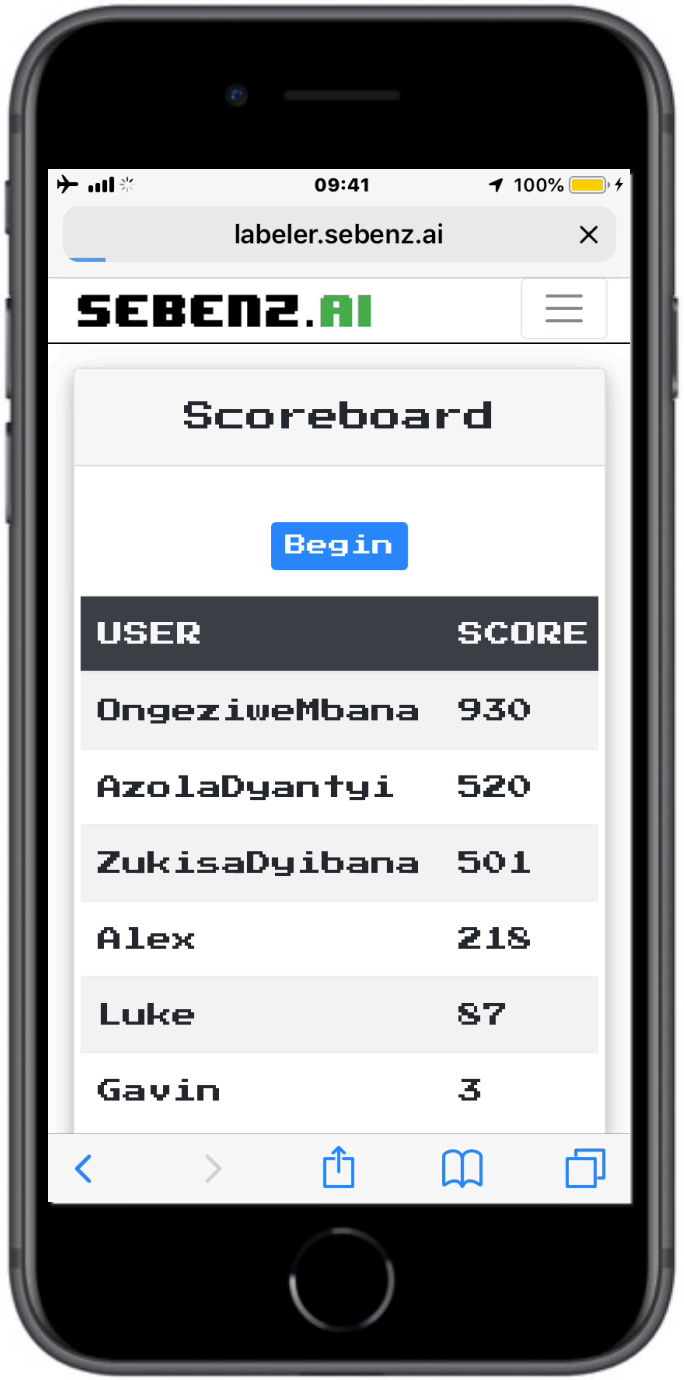
VERB

1 **work**

'Ngiyesaba mina ukusebenza ebusuku laphaya' — I am afraid to work over there at night

'Uthe uNkk. Sibongile Khoza ubesesikoleni esebenza ngesikhathi ehlaselwa yilo mfundi' — He said Mrs Sibongile Khoza was still busy working at school when she was attacked by the student


SEBEN2.AI


09:41
100%
labeler.sebenz.ai
SEBENZ.AI
Scoreboard
Begin
USER
SCORE
OngeziweMbana
930
AzolaDyantyi
520
ZukisaDyibana
501
Alex
218
Luke
87
Gavin
3

ARTIFICIAL INTELLIGENCE CREATING INSTEAD OF TAKING JOBS

https://clients.sebenz.ai/

SEBENZ.AI

projects docs log out

IMAGES


image classification


object detection


image description


polygon annotation


image segmentation


content moderation


document digitization

TEXT


text classification

text translation


entity extraction


topic detection

question answering

AUDIO


audio transcription


emotion detection

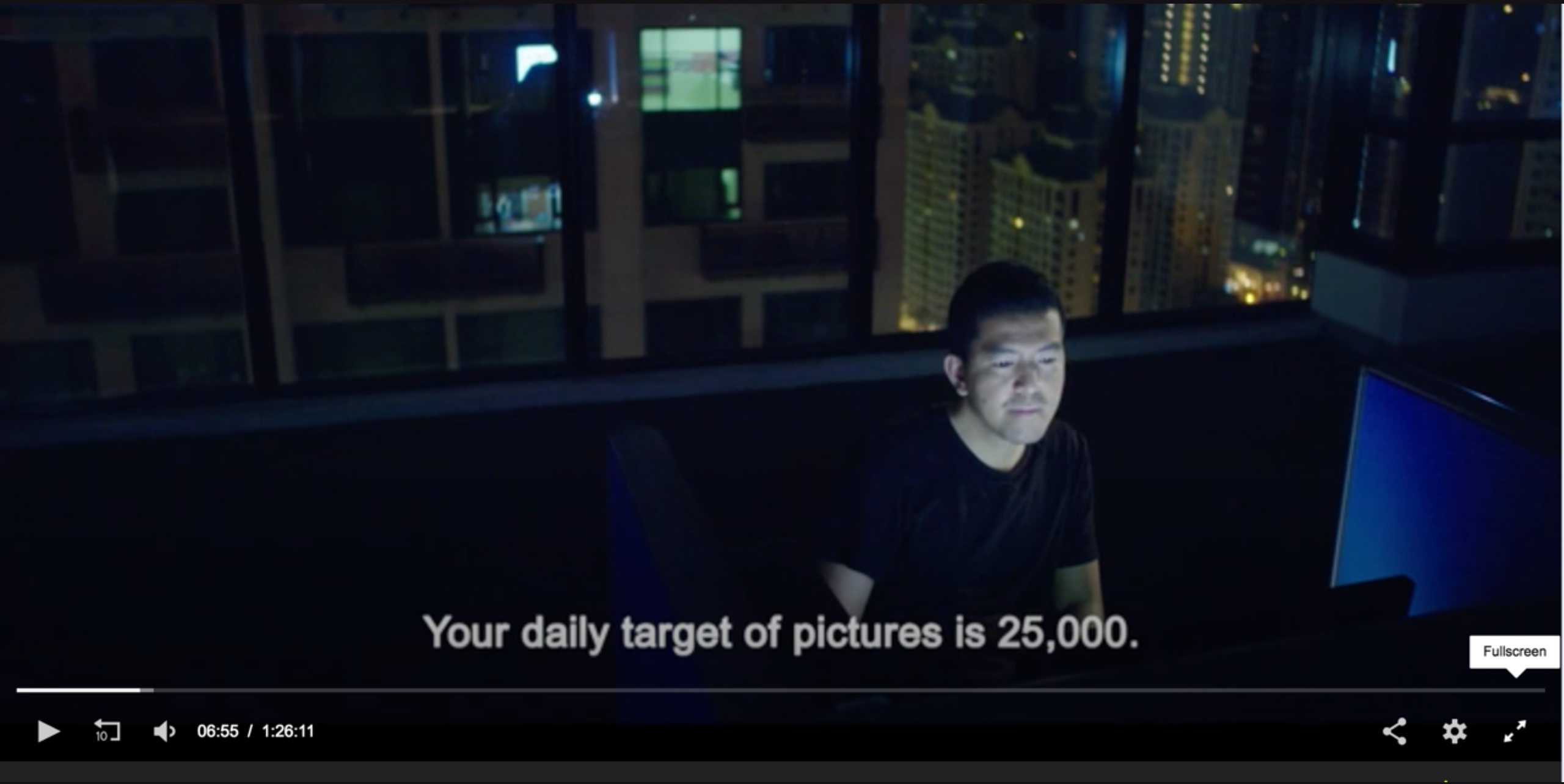

speaker identification

SEBEN2.AI

Score: 8

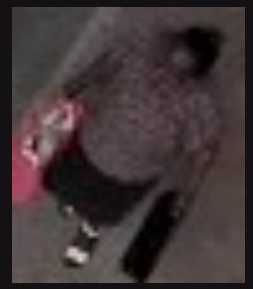

INDEPENDENT
LENS
THE CLEANERS
Monday NOV 12 10/9c
PBS
SUPPORT PROVIDED BY: ITVS, CPB, PBS, THE JOHN D. AND CATHERINE T. MACARTHUR
FOUNDATION, WYNCOTE FOUNDATION, AND THE NATIONAL ENDOWMENT FOR THE ARTS.

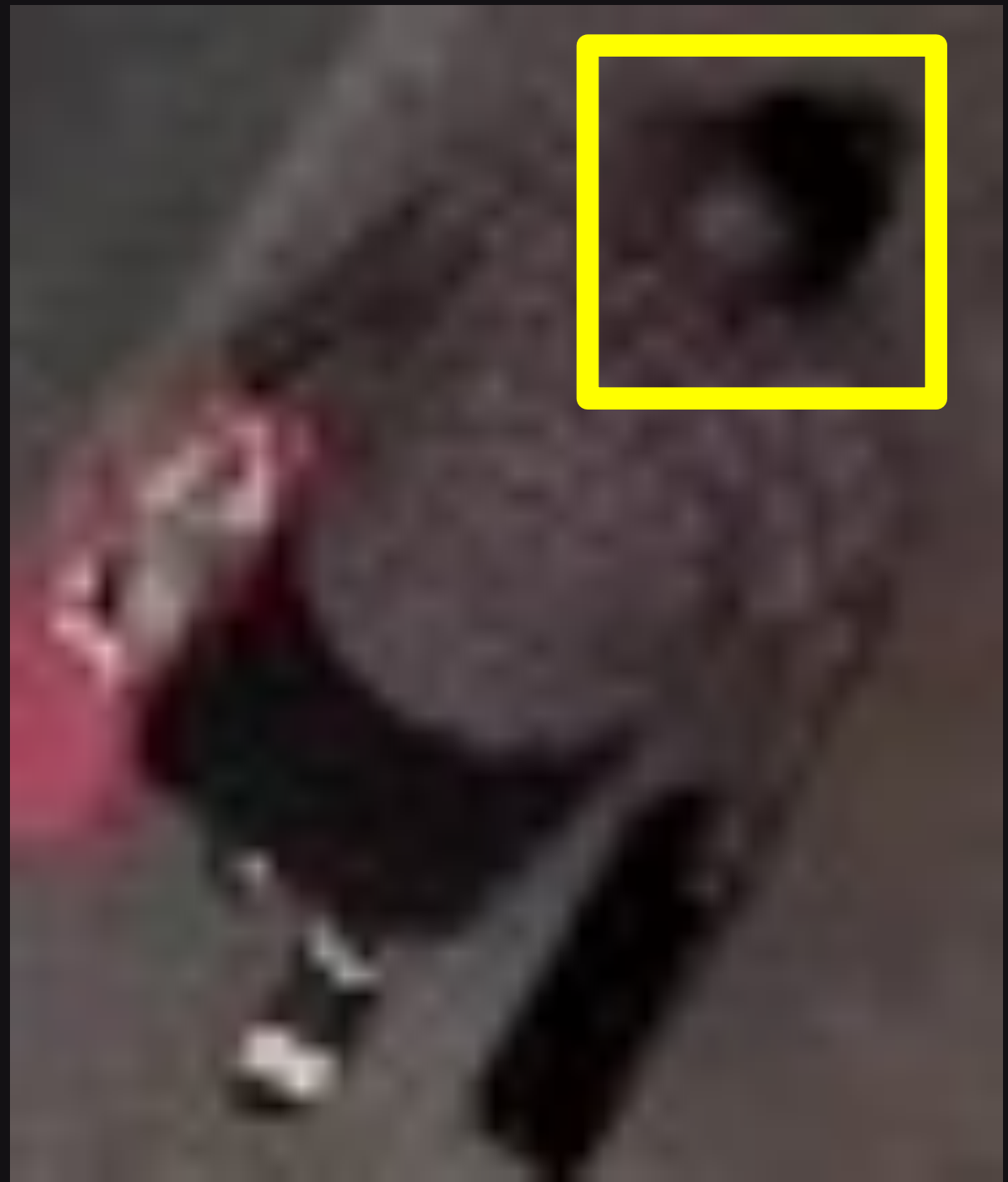

Press esc to exit full screen
00:00 / 03:09


Your daily target of pictures is 25,000.
Fullscreen
06:55 / 1:26:11

TRUSTLESS


2018-Nov-03 04:30:51.332 PM (SAST)
15
SVV-2960 - Port 1





ACCURACY?

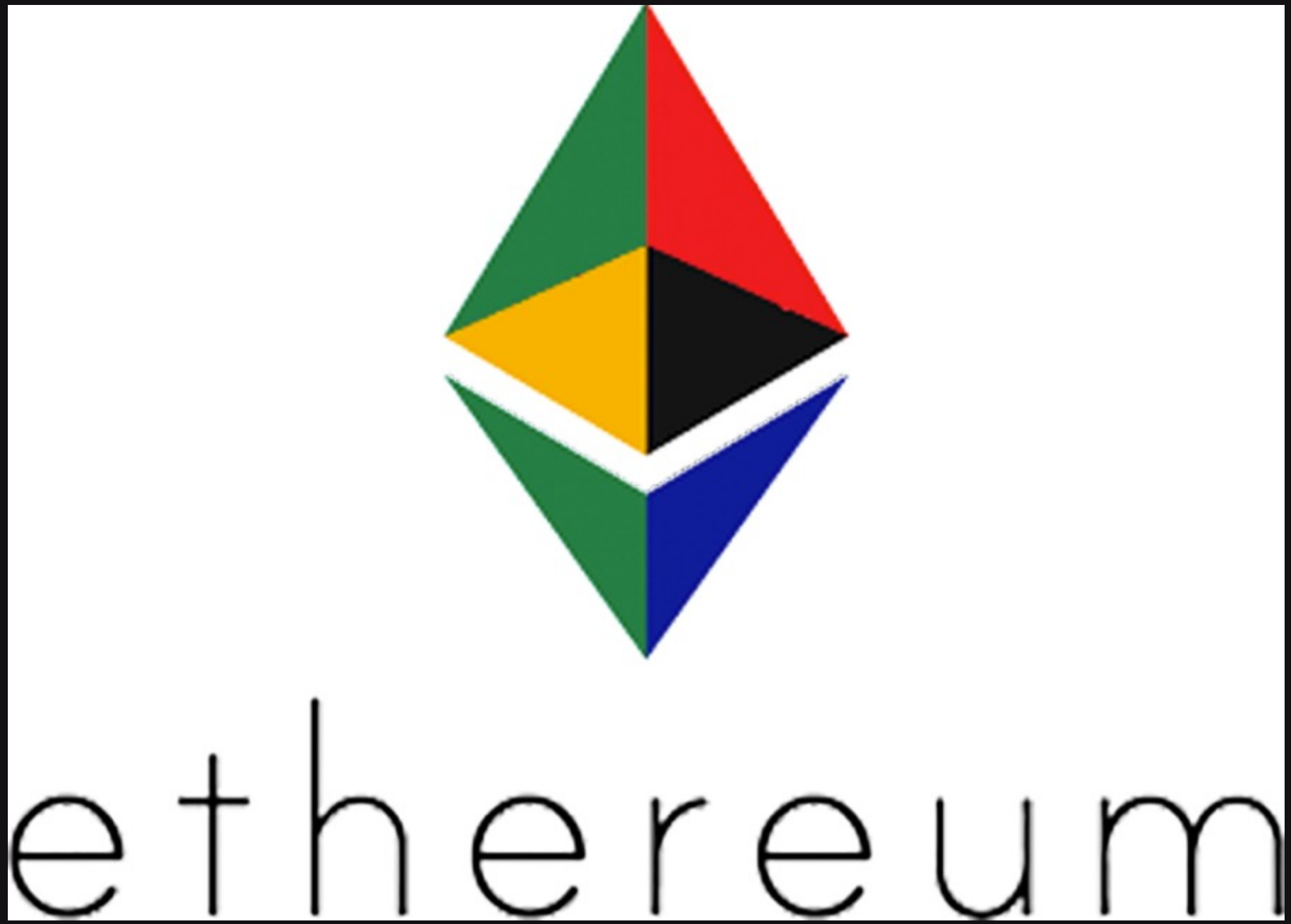

2018-NOV-06 06:20:15:245 PM (SAST)
15

$$\text{IOU} = \frac{\text{area of overlap}}{\text{area of union}}$$


GAMIFIED

SKILL-WEIGHTED CONSENSUS BETWEEN WORKERS


LIVES!


ethereum

ETHCapeTown @ETHCapeTown · Apr 21

Our final winners in no particular order are:

Wildcards
Snap
dTok
Sebenzai

WELL DONE EVERYONE!!! 💥🤯
More details on each to follow!
@ETHGlobal / @LinumLabs

OUR MISSION:

CREATE

1 MILLION JOBS

IN AFRICA!

SEBENZ.AI

plz get involved – email me! :)

alex @ sebenz.ai