# Deep Learning Architectures for Image Classification and Object Detection

**Nelson Correa,** ***Ph.D.***

*Data Science, Machine Learning and Natural Language Processing*

https://linkedin.com/in/ncorrea (https://linkedin.com/in/ncorrea)

**Palm Beach Data Science Meetup (https://www.meetup.com/Palm-Beach-Data-Meetup/events/262988444/)** West Palm Beach, FL

## Outline

1. Introduction: Computer vision and deep neural networks
2. Computer vision tasks and datasets
3. Deep neural network advances since 2011
4. MLP and CNN: Deep neural networks (MNIST)
5. Image classification with VGG16 (ImageNet classifier)
6. Object detection with YOLOv3 (MS-COCO object detector)
7. New developments and conclusion

Aiming for 45 minutes of presentation and 10 - 15 minutes of Q&A.

# Deep Learning Architectures for Image Classification and Object Detection

**Nelson Correa, *Ph.D.***
*Data Science, Machine Learning and Natural Language Processing*
https://linkedin.com/in/ncorrea (https://linkedin.com/in/ncorrea) @nelscorrea (https://twitter.com/nelscorrea)

## Abstract

Object detection is a task in computer vision with many practical applications that can now be achieved with super-human levels of performance on selected benchmarks using deep neural networks. In this talk we define the *object detection* task and present J. Redmon's YOLO (You Only Look Once) V3 deep neural network architecture. As preliminaries to object detection and YOLOv3, we first describe image classification on the Pascal VOC and ImageNet benchmark datasets, and introduce a series of deep learning neural network architectures that include the multilayer perceptron (MLP), convolutional neural networks (CNNs), and other networks with dystopian names such as AlexNet, GoogLeNet/Inception, VGG16, ResNet, and Region-CNN (R-CNN). We conclude with note of recent developments, including *capsule networks* (CapNets) by G. Hinton and deep networks with visual feedback. Slides and notebooks with code will be available after the talk.

## Speaker Bio

Nelson Correa is a data scientist and machine learning consultant based in West Palm Beach. He has a Ph.D. in Electrical Engineering and over 25 years of experience in natural language processing at three startup companies and IBM Research. He has over 30 technical publications and three U.S. patents. His current interest is in developing connections between deep learning and symbolic models for natural language processing and perception.

# 1. Introduction: A brief chronology

## Computer vision and deep neural networks (1950 - 2000)

1950s - R. Rosenblatt, W. McCoullogh, W. Pitts (perceptron); Hubel, Weisel (cats)

1960s - MIT AI: Minsky, Waltz, Winston, Marr; AI, Blocks World, Robotics

1970s - MIT AI & computer vision; E. Harth, Alopex; G. Hinton, connectionist networks

1980s - Symbolic AI & computer vision; Connectionism

- Rumelhart *et al.*, Parallel Distributed Processing (PDP)
- G. Hinton & others, MLP, Backpropagation
- Y. LeCun & others, Convolutional neural networks (CNN)

1990s - Pattern recognition (PAMI), machine learning

## Computer vision and deep neural networks (2000 - present)

2000s - Deep learning, OpenCV

- Bengio; neural language models
- Hinton, LeCun, Bengio & others; training deep networks
- G. Bradsky, OpenCV
- Stanford, Toronto, MILO, Oxford; Datasets and benchmarks (Pascal VOC, CIFAR)

2010-2019 - Rapid progress in deep learning

- Google, Microsoft, Facebook; Benchmarks, datasets (ILSVCR, MS-COCO)
- AlexNet on ImageNet/ILSVRC, 2012; Bengio, ReLU
- Deep network modules, architectures, and toolkits
- Hinton, LeCun, Bengio (ACM Turing Award, 2019)

# 2. Computer vision tasks and datasets

*"The classical problem in computer vision, image processing, and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. "*
(https://en.wikipedia.org/wiki/Computer_vision#Recognition (https://en.wikipedia.org/wiki/Computer_vision#Recognition))

- The terminology varies and includes the terms **recognition**, **classification**, **identification** and **detection** when used to describe computer vision tasks.
- A finite set of **labels** or classes (e.g., cat, dog, person, cup, lawn) is assumed to be associated with images, objects and other features of images.
- The denotations of the *labels* may be exclusive (disjoint), or they may overlap and define a hierarchy (e.g., of concepts).

### NOTES:

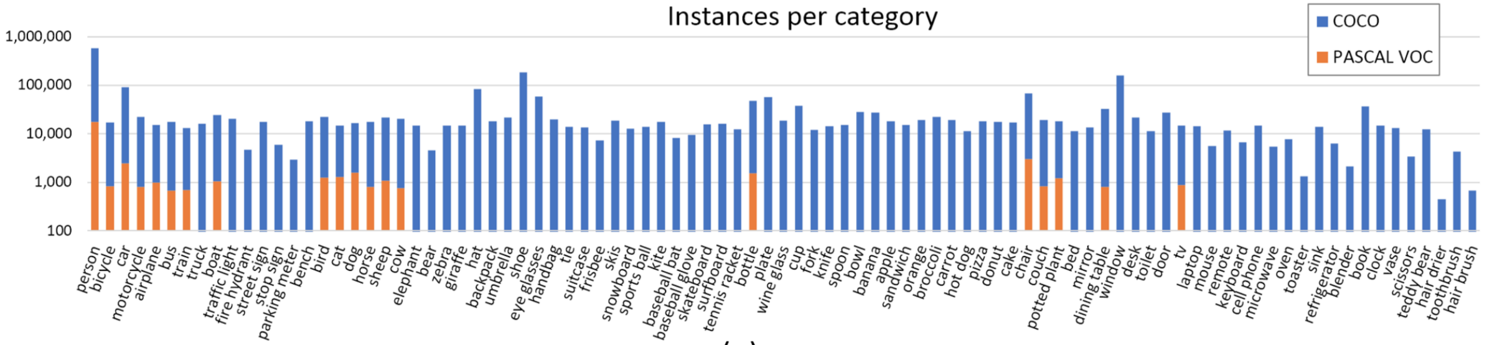
- Instead of finite, symbolic *labels*, we may define image or class ***attributes*** (in a discrete or continuous space).
- The *classical problem* is assumed to be a **supervised** learning problem - correct label or labels for an image given *by example* (an *<image, labels>* training set).

## 2.1 Image Classification (logistic regression)

- Given an **image** and a set of **labels** or classes, determine what *labels* should be assigned to the image.
- The problem may be **multi-class** (multinomial classification; more than two classes)
- The problem may be **multi-label** (each image may assigned multiple labels)

#### Pascal VOC (20 classes) - MS COCO (80 classes):

- Person: person
- Animal: bird, cat, cow, dog, horse, sheep
- Vehicle: aeroplane, bicycle, boat, bus, car, motorbike, train
- Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor

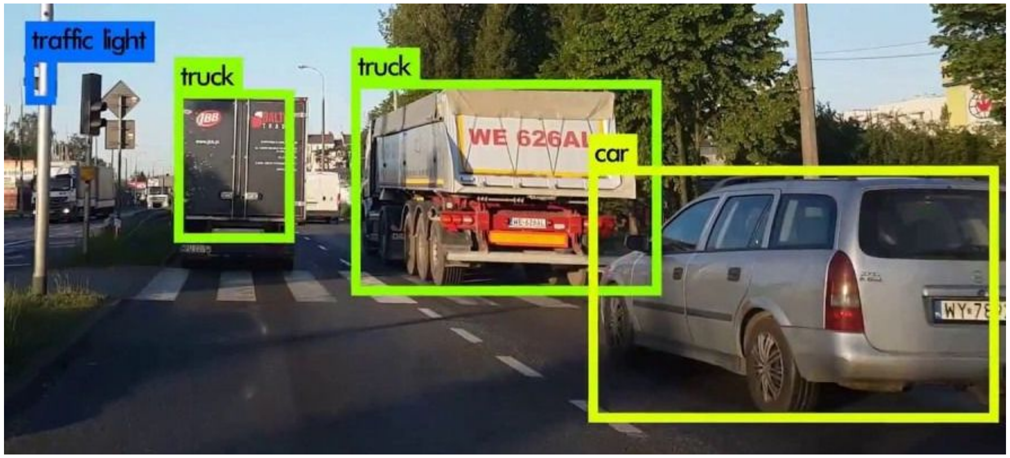


(a)

## 2.2 Object Detection (logistic+linear regression)

- Given an **image** and a set of **object labels**,
  - (i) determine what **labels** should be assigned to the image (objects in image)
  - (ii) indicate the **bounding box** (location) of each object in the image
  - (iii) provide a confidence score (**objectness**) for each label selected
- *Bounding box* represented by four coordinates *(x, y, width, height)*, relative to image.

### Object Bounding Boxes



## 2.3 Other computer vision tasks

- Character recognition
- Object localization
- Object segmentation
- Object/Person identification
- Object/Person verification

## 2.4 Multi-modal CV tasks (vision and language)

- Image captioning, description
- Visual/Image Question Answering (VQA)

## 2.5 Computer vision datasets

### Datasets

- **MNIST** (60,000 28x28x1 handwritten digits 0-9; AT&T Bell Labs; 1989-1998)
- **Pascal VOC** (12,500 images; 20 classes; U. Oxford; 2005-2012)
- **CIFAR** (60,000 32x32x3 images; 10 or 100 classes; U. Toronto; 2009)
- **MS-COCO** (200,000 images; 80 classes; Microsoft; 2014-2019)
- **ImageNet** (14M images; 21,000 classes; Google/Stanford; 2010-2019)

... other

### Image types

- Iconic objects and scenes
- Non-iconics object and scenes (common objects in context)

"*ImageNet is* ***an image database organized according to the WordNet hierarchy*** *(currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images.*"

- ImageNet (ILSVRC): http://image-net.org (http://image-net.org)
- WordNet: http://wordnet.princeton.edu/ (http://wordnet.princeton.edu/)
- *ImageNet Large Scale Visual Recognition Challenge*, Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei, International Journal of Computer Vision, 2015. https://arxiv.org/abs/1409.0575 (https://arxiv.org/abs/1409.0575)

## ImageNet Summary and Statistics (updated on April 30, 2010)

- Total number of non-empty synsets: 21,841
- Total number of images: 14,197,122
- Number of images with bounding box annotations: 1,034,908
- Number of synsets with SIFT features: 1000
- Number of images with SIFT features: 1.2 million

Source: http://image-net.org/about-stats (http://image-net.org/about-stats)

# 3. Deep neural network advances since 2011

## 3.1 Deep neural networks pre-2011

- Perceptron (trainable; direct solutions or gradient descent). Too simple a learning function (cannot learn XOR; Minsky and Papert, 1969)
- Multi-Layer Perceptron (MLP) and deep networks (backpropagation ~ chain rule)
    - ... but deep MLP not practically trainable (vanishing gradients, complexity)
    - simple architecture: not hierarchical; activation function; feed-forward
        - no feedback; no "attention" mechanisms
- Convolutional neural networks (CNNs) with a few layers (LeCun, 1989)
    - CNNs provide *hierarchy*, an advance over fully-connected layers (MLP)
        - spatial and visual feature/concept organization
    - But deep CNNs were not yet practical (i.e., trainable) in 1989
- Parallel Distributed Processing
    - CV as combined symbolic AI and pattern recognition
    - IEEE PAMI, OpenCV

## 3.2 Deep neural network advances since 2011

- Improved deep network training (2011 ILSVRC visual recognition challenge)
  - More effective neuron activation functions
    - (from *sigmoid* to *tanh, reLU, leaky reLU*)
  - New training methods (SGD, batch/mini-batch SGD; scaling; optimizers)
  - Regularization (network weight rescaling; dropout; data augmentation)
  - Reusable transfer learning
- Improved deep network models and modules (2011 - present)
  - Various new neural network modules (Inception, Xception, Capsules)
  - Residual neural networks (ResNet)
  - Deep, large network architectures for computer vision (LeNet, GoogLeNet/Inception, VGG, R-CNN, YOLO)
  - Attention, feedback (Feedback-CNN) and capsules (CapsNet)
- Large image datasets, CV tasks
  - Image tasks and datasets: CIFAR, Pascal VOC, MS COCO, ImageNet
  - Transfer learning on deep networks trained on the large datasets

# 4. MLP and CNN: Deep neural networks (MNIST)

### MNIST dataset

- *The MNIST database of handwritten digits*, Yann LeCun, Corinna Cortes, and Christopher JC Burges, 1998
- LeCun site (http://yann.lecun.com/exdb/mnist/)

### MNIST MLP

- Input (28, 28) - Dense-256-relu - Dense-10-softmax
- Input (28, 28) - Dense-128-relu - Dense-128-relu - Dense-10-softmax

### MNIST CNN

- **Feature extractor**: Input (28, 28), Conv2D-32, Conv2D-64-relu, Conv2D-64-relu
    - **Output classifier**: Flatten, Dense-128-relu, Dense-10-softmax

See notebook: MNIST_Digit_Classification.ipynb (./MNIST_Digit_Classification.ipynb) - html (./MNIST_Digit_Classification.html)

# 4.1 MNIST State-of-the-Art (SOTA)

## MNIST Test accuracy stands at 99.83% (in 2019)

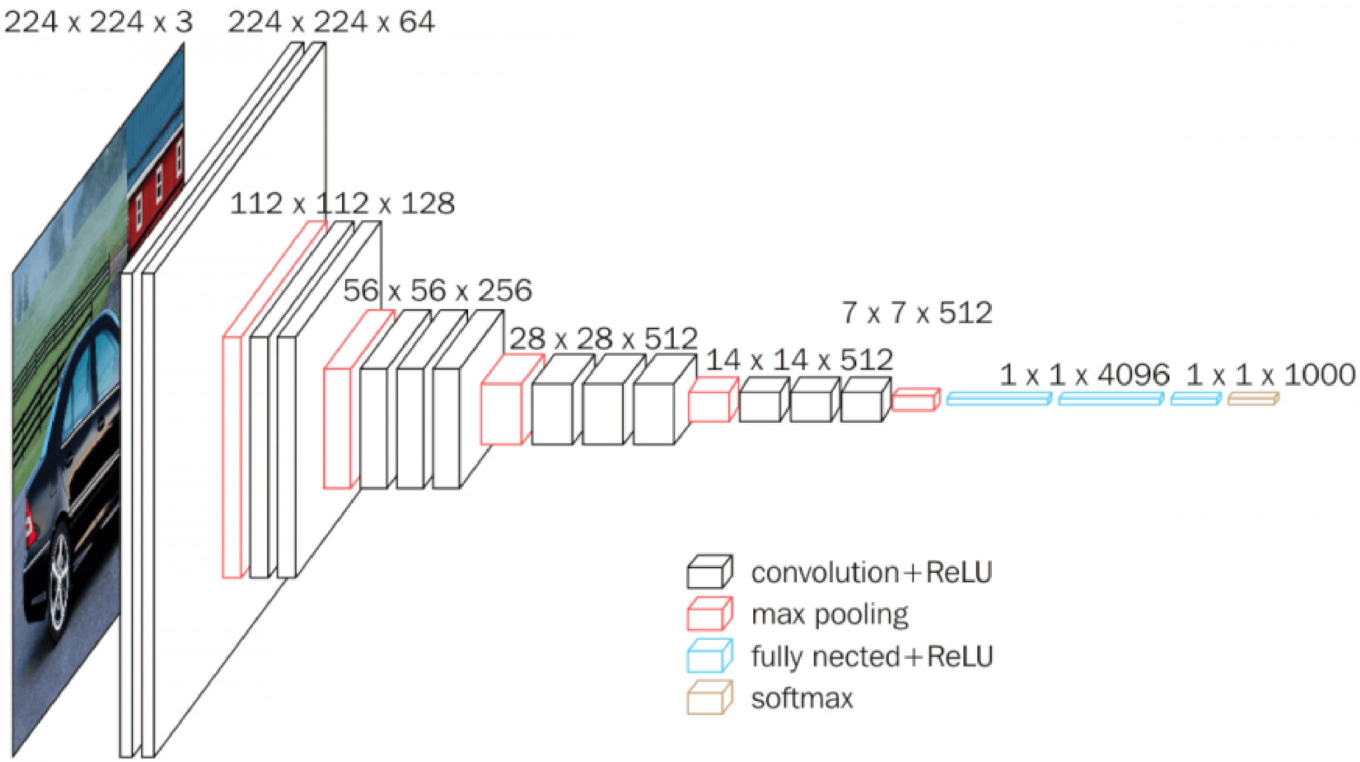
### Record (03/2019)

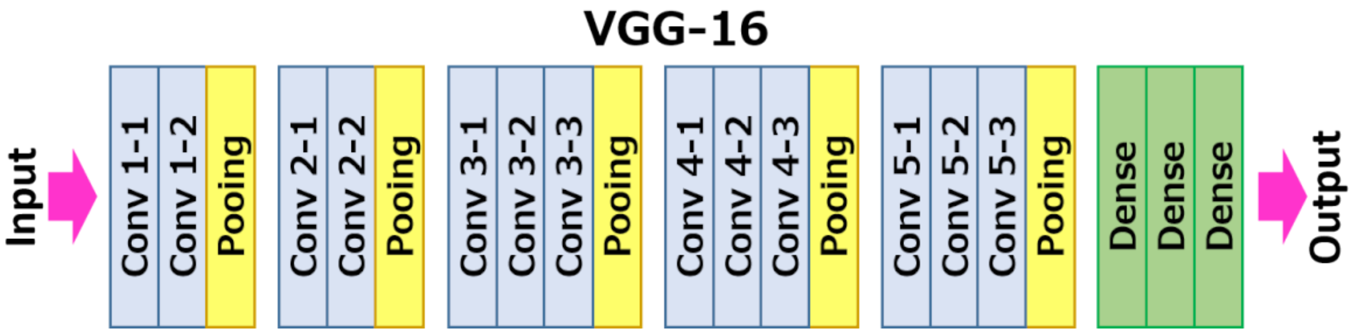
- Zhao et al., 2019, report absolute MNIST error rate reduction from previous best 0.21% (99.79% accuracy) to 0.17% (99.83% accuracy).
- *Capsule Networks with Max-Min Normalization*, Zhen Zhao, Ashley Kleinhans, Gursharan Sandhu, Ishan Patel, K. P. Unnikrishnan, 2019, https://arxiv.org/abs/1903.09662 (https://arxiv.org/abs/1903.09662).

### Others (2017 - 2018)

- *Regularization of neural networks using dropconnect*, Li Wan, Matthew D Zeiler, Sixin Zhang, Yann LeCun, and Rob Fergus, ICML 2013. 99.79% accuracy (0.39% error rate; 0.21% with ensembling)
- *Dynamic Routing Between Capsules*, Sara Sabour, Nicholas Frosst, Geoffrey E Hinton, 2017 (https://arxiv.org/abs/1710.09829 (https://arxiv.org/abs/1710.09829)) 99.75% accuracy (0.25% error rate)
- Simple 3-layer CNN above (total params, 130,890)
  without any training regularization: `acc: 99.15%`

# 5. VGG16 Image classification (ImageNet)





# 5.1 Image classification with VGG16

See notebook: VGG16_Image_Classification.ipynb (./VGG16_Image_Classification.ipynb) - html (./VGG16_Image_Classification.html)

## Image classification examples and issues with VGG16 in Keras

Image classification task

- Multiclass; Multilabel

VGG16 image classification architecture

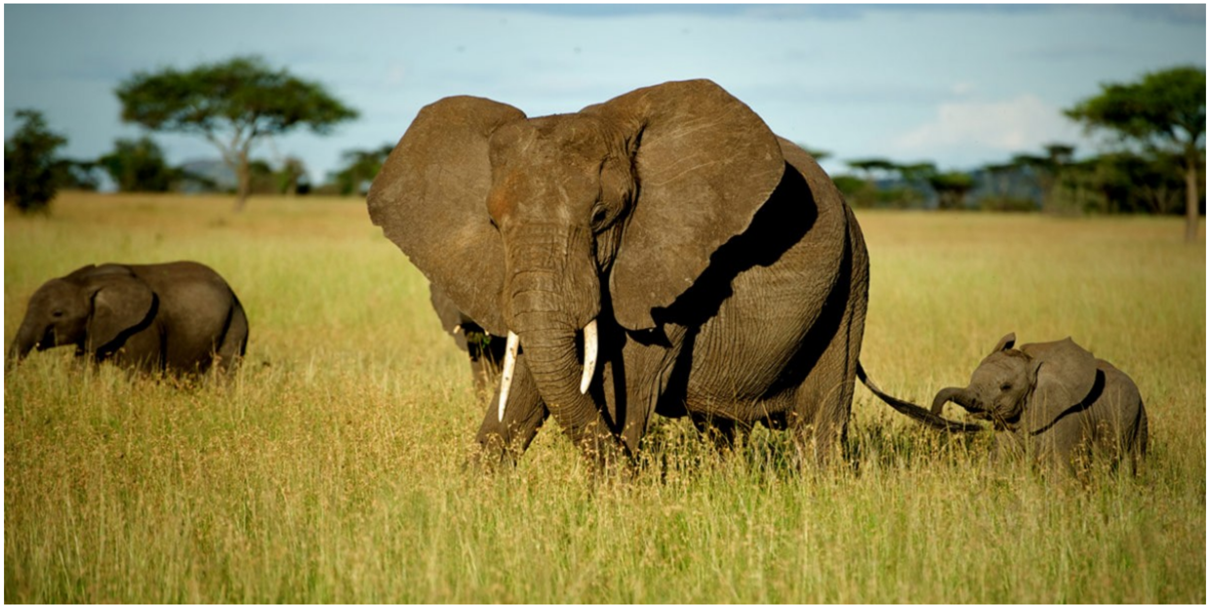
- Deep CNN with output classifier
- 16 trainable layers (13 convolutional; 3 FC classifier); 23 total layers

Pretrained and custom VGG16 image classifiers

- Pretrained VGG16: 1000 ImageNet classes
- custom trained output classifier: Dogs vs. cats

# 5.2 VGG16 Example - African elephant

## African elephant (Flickr)

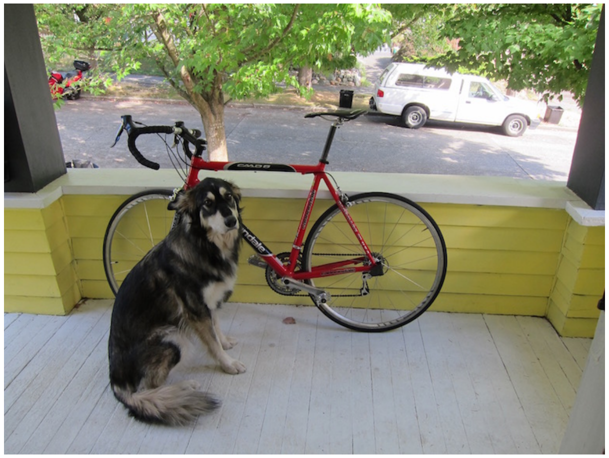


### Top-5 Predictions

```
n02504458 - African_elephant 69.72%
n01871265 - tusker       19.20%
n02504013 - Indian_elephant 6.60%
n02410509 - bison        3.56%
n02437312 - Arabian_camel 0.43%
```

# 5.2 VGG16 Example - African elephant

## Dog, Bike and Truck (J. Redmon)

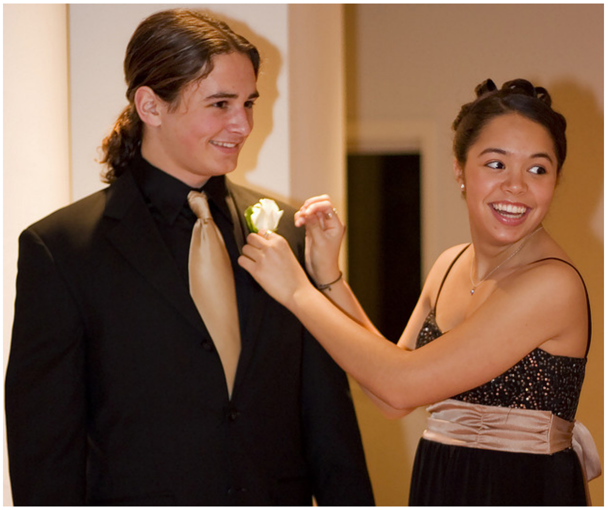


### Top-5 Predictions

```
n02110063 - malamute     32.37%
n02110185 - Siberian_husky 21.75%
n02109961 - Eskimo_dog   15.27%
n03218198 - dogsled      5.32%
n02106166 - Border_collie 4.22%
```

# 5.2 VGG16 Example - People

## Young man and woman (MS-COCO)



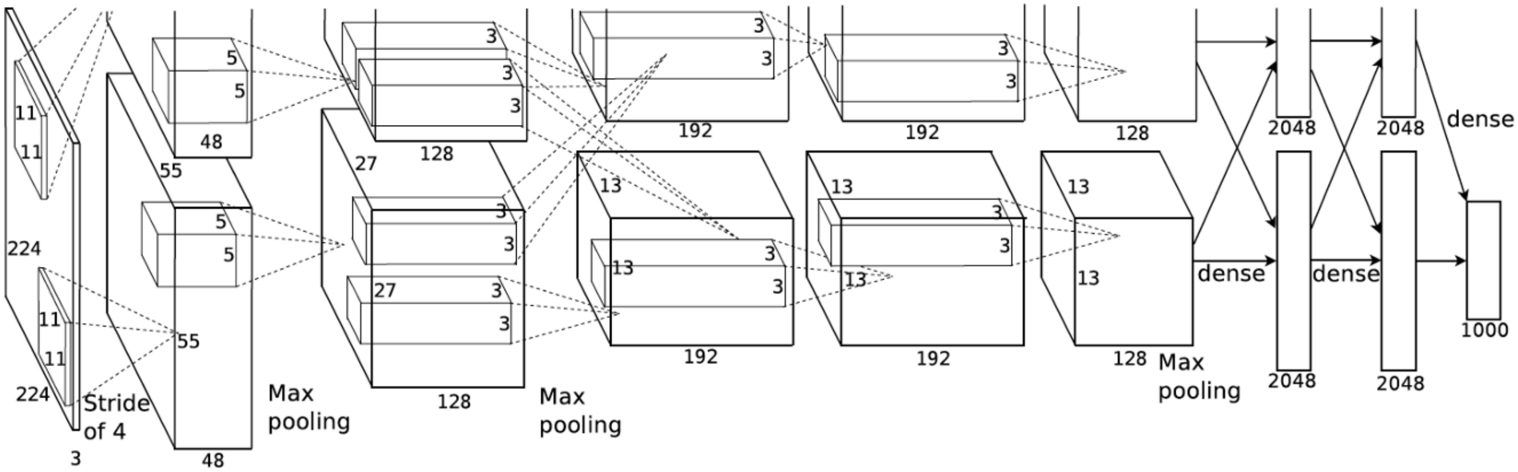
### Top-5 Predictions

```
n10148035 - groom        39.19%
n02883205 - bow_tie      27.01%
n04350905 - suit         11.94%
n03450230 - gown         8.47%
n03770439 - miniskirt    2.00%
```

# 5.3 Other image classification architectures

## AlexNet (Krishevzky *et al.*, 2012, ILSVRC)

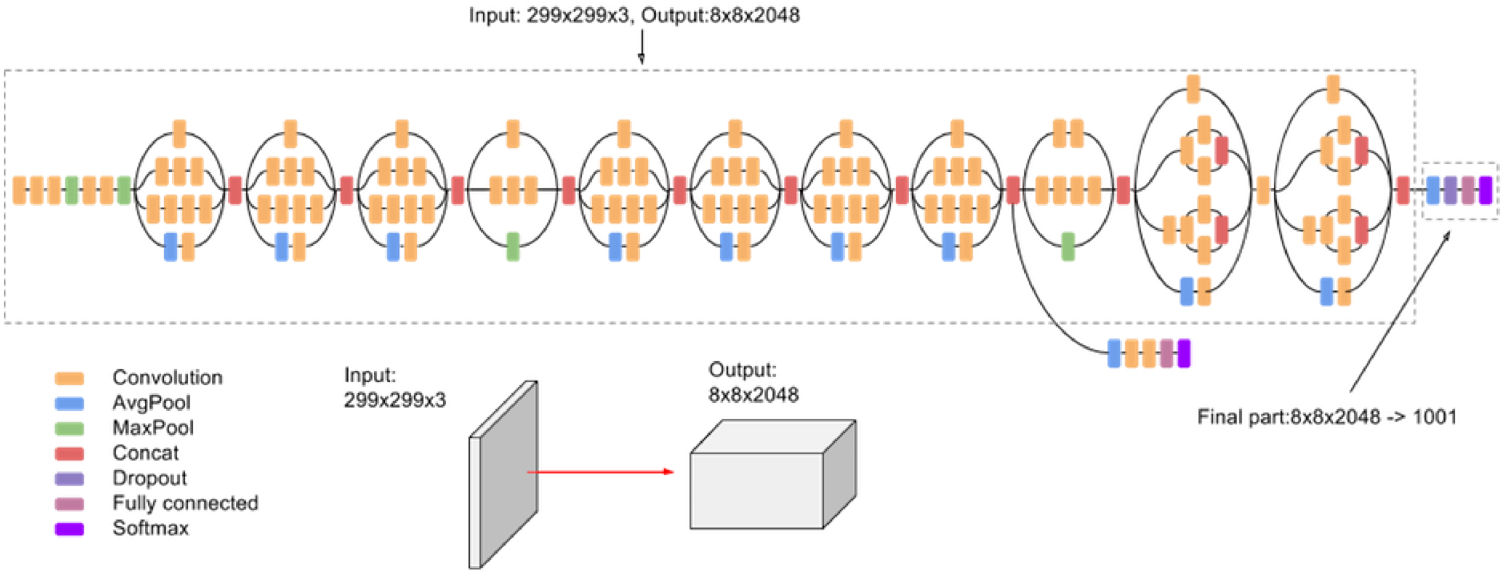
AlexNet_architecture_01.png



Input Conv2D Max-Pooling Dense

# 5.4 Other image classification architectures

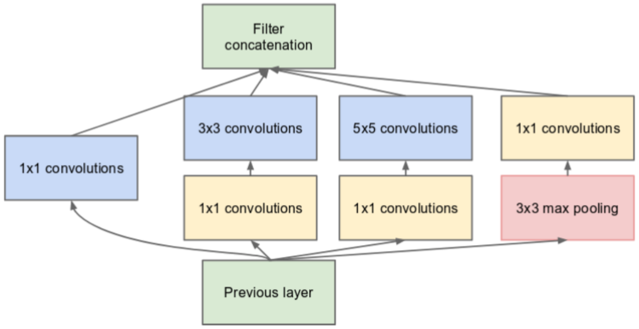
## Inception V3 (2015)



- https://cloud.google.com/tpu/docs/inception-v3-advanced (https://cloud.google.com/tpu/docs/inception-v3-advanced)

# 5.5 Deep learning architecture improvements

## InceptionV1 (2014), ResNet, ResNeXt (2016)



(b) Inception module with dimension reductions

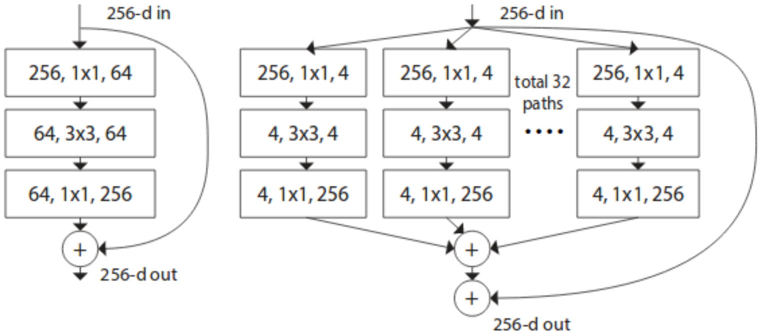


Figure 1. **Left**: A block of ResNet [14]. **Right**: A block of ResNeXt with cardinality = 32, with roughly the same complexity. A layer is shown as (# in channels, filter size, # out channels).

# 6. Object Detection

An extension of *image classification*.

## Image classification

**input image** -> **class probabilities** (multinomial logistic regression)

## Object localization and detection

(single object): **input image** ->

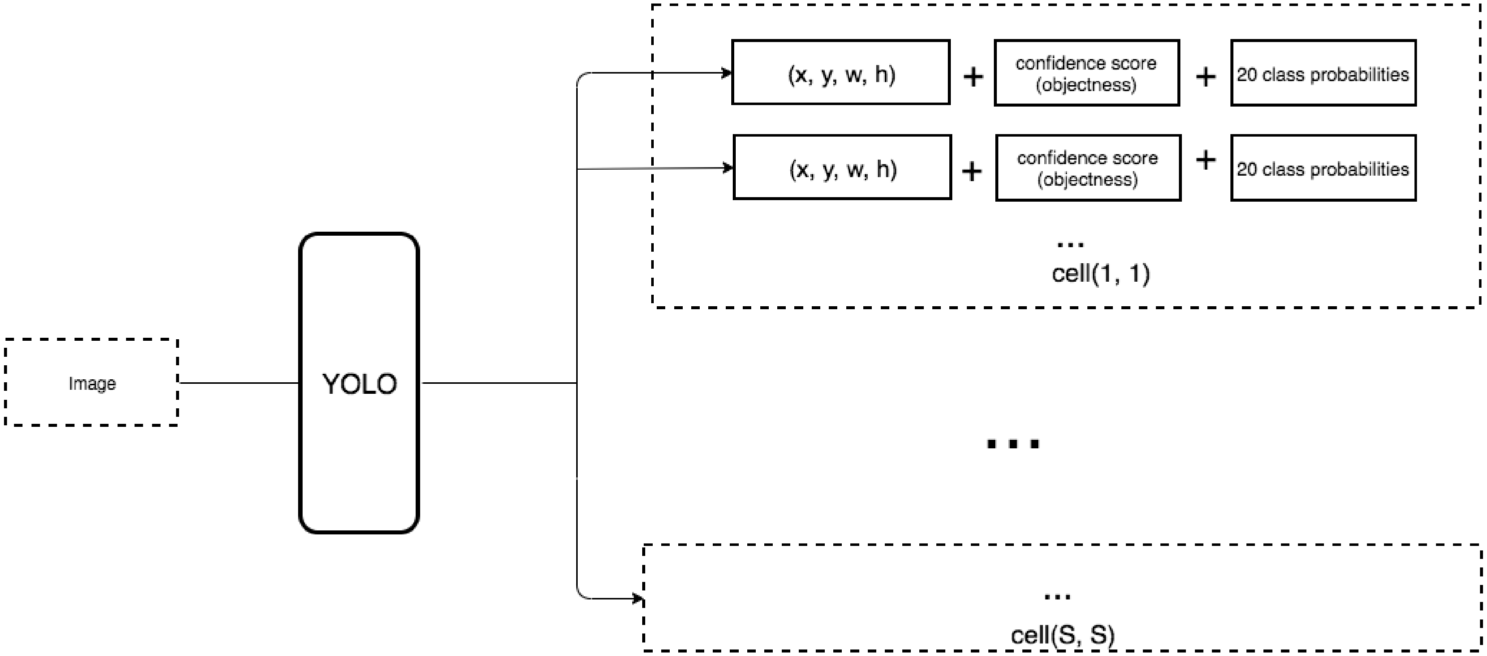
- **class probabilities** (multinomial logistic regression),
- **bounding box** (x, y, width, height)

(multiple objects at different scales): **input image** -> plural image grid cells

- **class scores/probabilities** (multinomial logistic regression),
- **objecness score** for cell,
- **plural bounding boxes** (x, y, width, height)

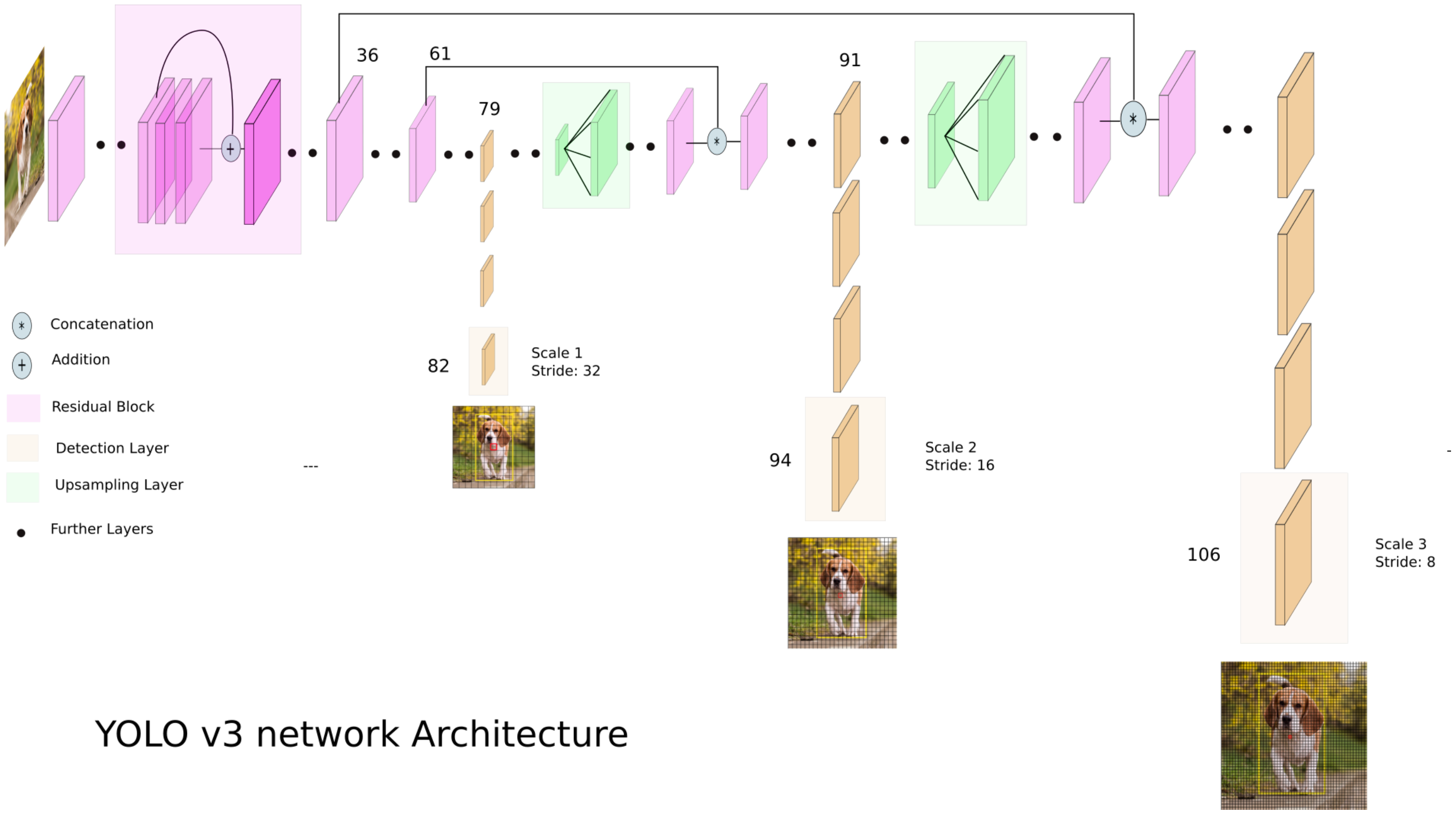
# 6.1 YOLOv1 Object detection

- YOLOv1 (Pascal VOC, 20 classes)
- YOLOv3 (MS-COCO, 80 classes) - Total yolov3.layers: 252 (Convolutional: 106)



See notebook: YOLOv3_Object_Detection.ipynb (./YOLOv3_Object_Detection.ipynb) - html (./YOLOv3_Object_Detection.html)

# 6.2 YOLOv3 Object detection (on MS-COCO)

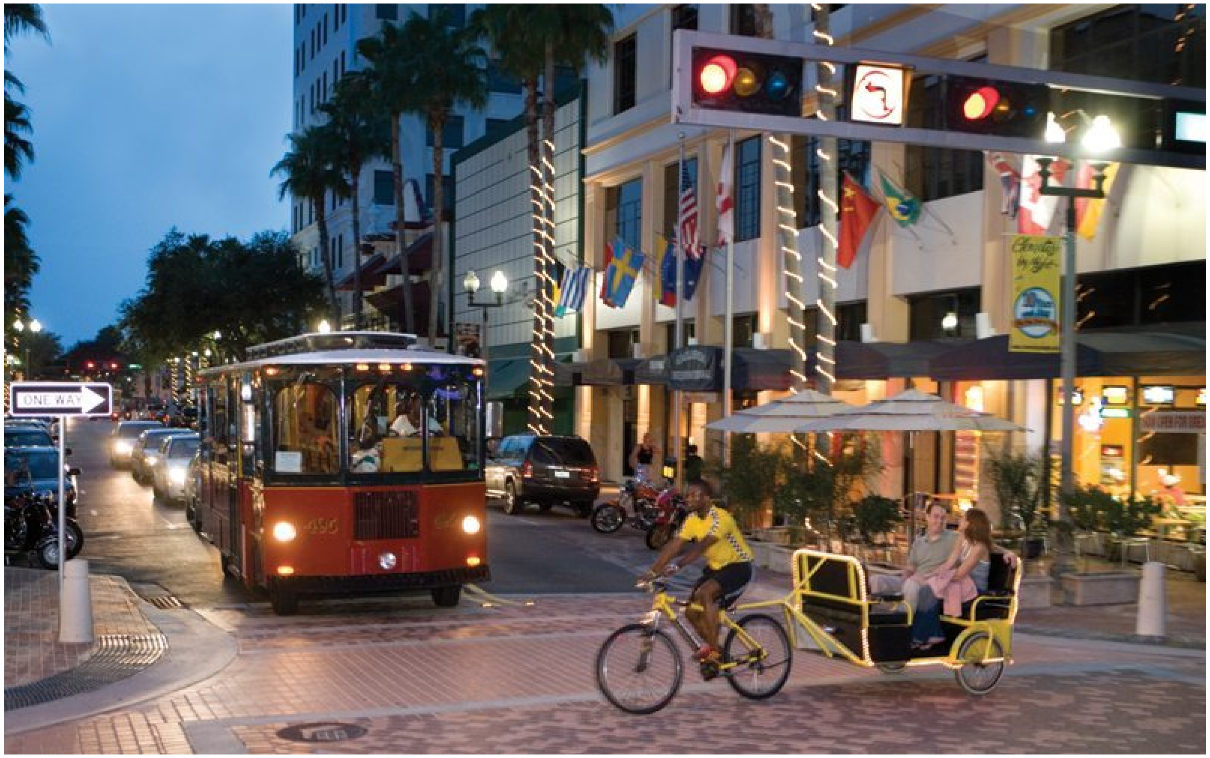


YOLO v3 network Architecture

**Source**: Ayoosh Kathuria, YOLOv3 **YOLO**: [https://pjreddie.com/yolo/ (https://pjreddie.com/yolo/)](https://pjreddie.com/yolo/)

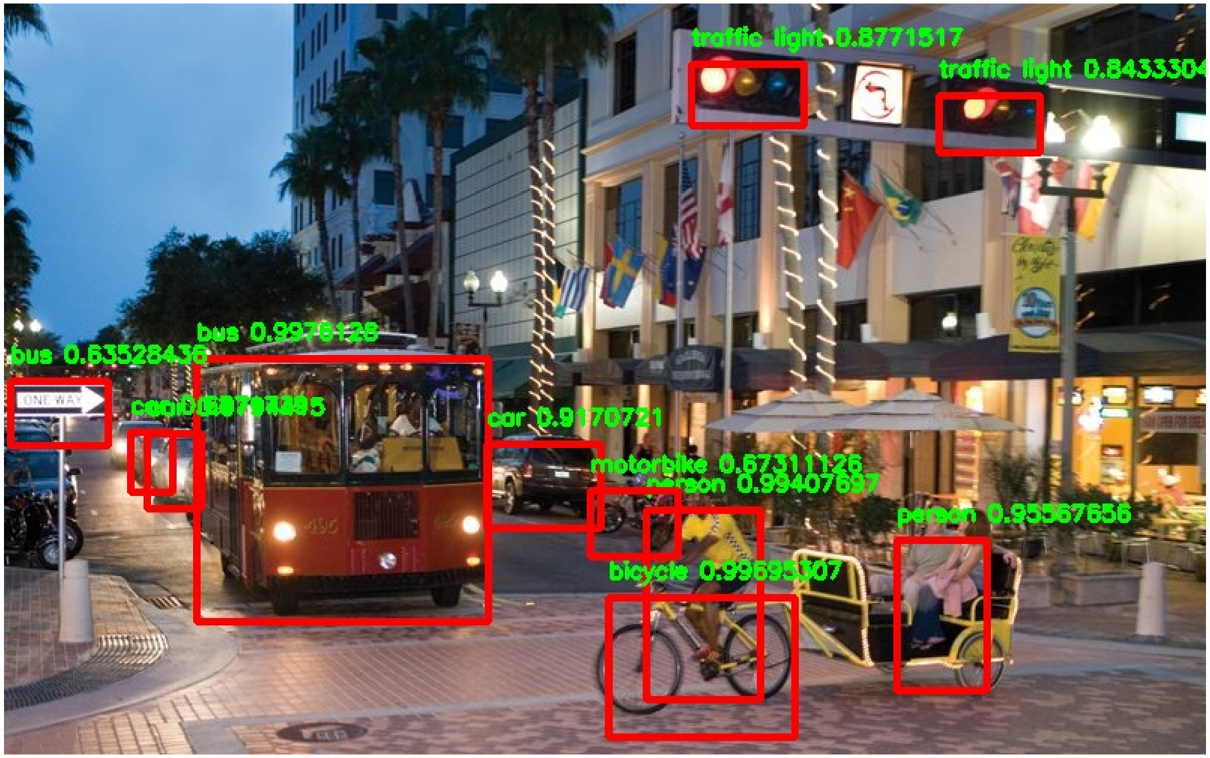
# 6.3 YOLOv3 Example - West Palm Beach

## Clematis Street



Evening on Clematis Street

# 6.3 YOLOv3 - Evening on Clematis Street



**Predictions on image (total objects: 11)**

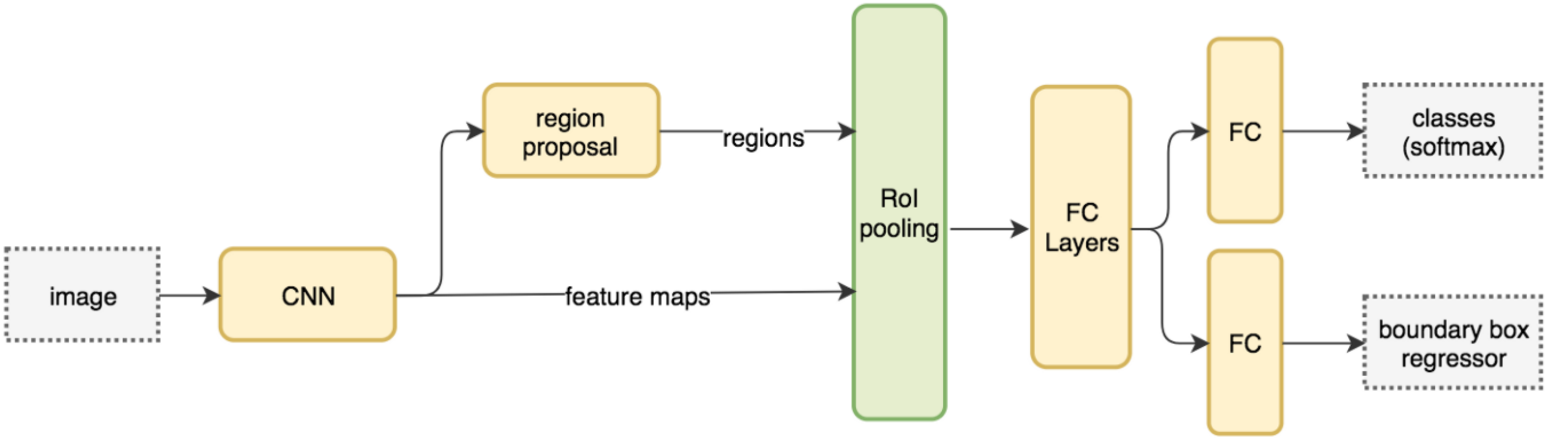
bus: 99.7612% person: 99.4076% person: 95.5676%
bicycle: 99.6953% traffic light: 87.7151% traffic light: 84.3330%
bus: 63.5284% car: 68.7173% car: 97.9441%
car: 91.7072% motorbike: 67.3111%

# 6.4 Other object detection architectures

## Region CNN (R-CNN, 2015 - 2018)

Series of models

- Ujlinjs, 2011
- R-CNN, 2014
- Fast R-CNN, 2014
- Faster R-CNN, 2014

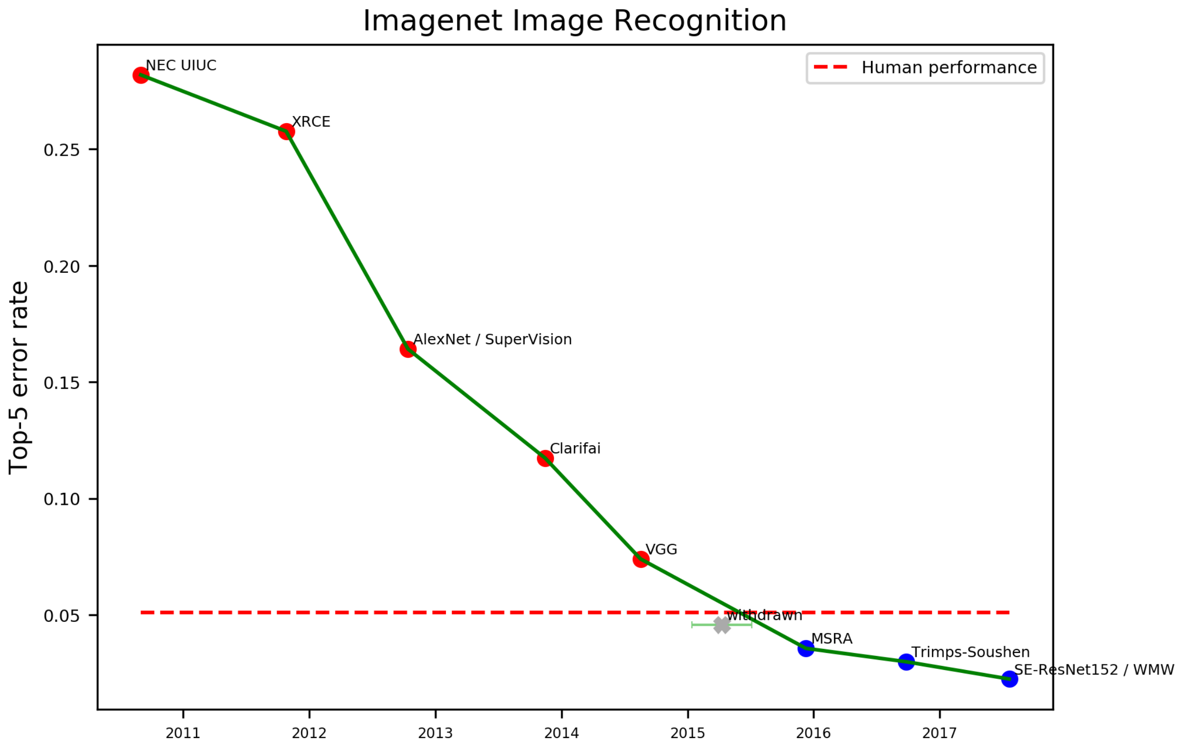


## SSD: Single-Shot Detector (20xx)

# 6.5 Benchmarks and Performance Numbers

## Image classification performance

ImageNet 2011 - 2017, human performance (95%) vs. super-human performance (98%) (Source: EFF AI Metrics (https://www.eff.org/ai/metrics))



## Object detection performance

Human performance

# 7. New neural models

## Feedback neural networks (inspired by the mamalian visual system)

*Role of feedback in mammalian vision: a new hypothesis and a computational model* P.S. Sastry, Shesha Shah, S. Singh, K.P. Unnikrishnan, Vision Research 39 (1999) 131–148, Elsevier.

*Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks*, Cao, Chunshui *et al.*, IEEE ICCV, 2015. pdf (https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Cao_Look_and_Think_ICCV_2015_paper.pdf) github (https://github.com/caochunshui/Feedback-CNN) IEEE PAMI (https://www.computer.org/csdl/journal/tp/2019/07/08370896/13rRUwdIOTs)

## Capsule networks

*Dynamic Routing Between Capsules*, Sara Sabour, Nicholas Frosst, Geoffrey E Hinton, 2017 (https://arxiv.org/abs/1710.09829 (https://arxiv.org/abs/1710.09829)) (https://github.com/XifengGuo/CapsNet-Keras (https://github.com/XifengGuo/CapsNet-Keras))

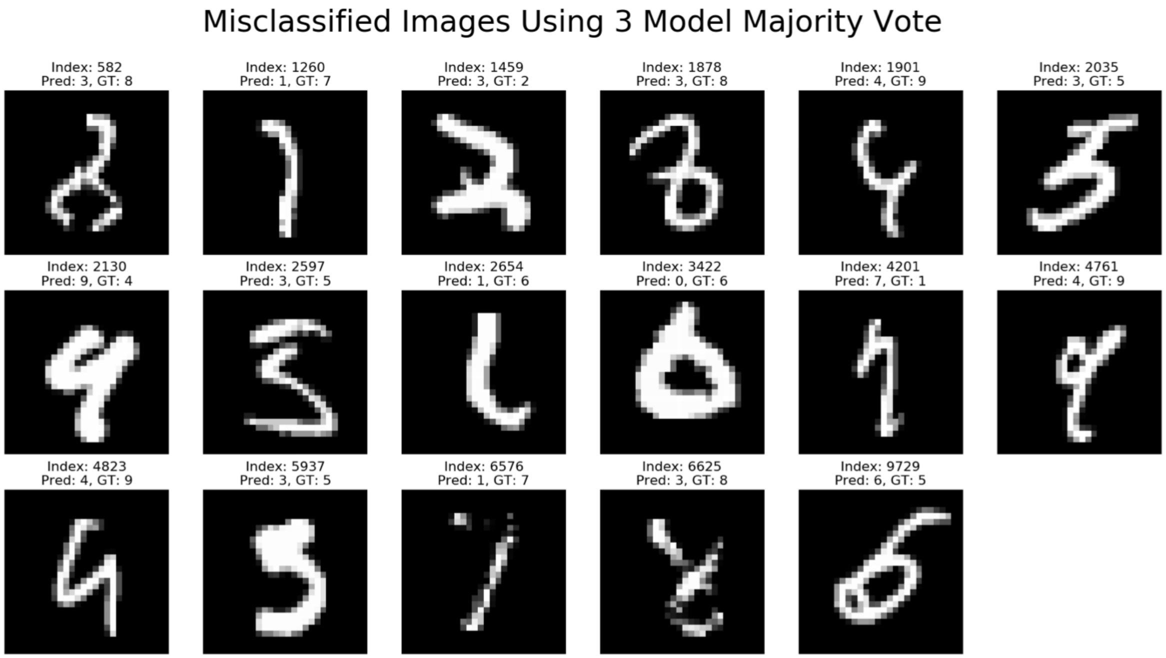
Absolute MNIST error rate reduction from 0.21% (99.79% accuracy) to 0.17% (99.83% accuracy), Zhao *et al.*, 2019.

*Capsule Networks with Max-Min Normalization*, Zhen Zhao, Ashley Kleinhans, Gursharan Sandhu, Ishan Patel, K. P. Unnikrishnan, 2019, (https://arxiv.org/abs/1903.09662 (https://arxiv.org/abs/1903.09662))

**NOTE**: After 20 years, **MNIST Test** can no longer be considered a proper *test* set (it is instead a *validation* set)

# CapsNet with Max-Min: MNIST Test set errors

Misclassified MNIST images using 3-model majority vote from CapsNets trained using Max-Min normalization



- Total of 17 digit errors in 10,000 digit test set (99,83% test accuracy)
- MNIST digit index 6576 - Ground Truth: "7"
- current deep networks recognize as "1"; no human makes such error

Source: *Capsule Networks with Max-Min Normalization*, Zhao et al., 2019 (https://arxiv.org/abs/1903.09662)

# New neural modules and networks for vision, cont'd

## Feedback neural networks

Deep networks with feedback from later layers inspired by the mamalian visual system.

- *The Inversion of Sensory Processing by Feedback Pathways: A Model of Visual Cognitive Functions*, E. Harth; K. P. Unnikrishnan; A. S. Pandya, 1987, Science, New Series, Vol. 237, No. 4811. (Jul. 10, 1987), pp. 184-187.
- *Role of feedback in mammalian vision: a new hypothesis and a computational model* P.S. Sastry, Shesha Shah, S. Singh, K.P. Unnikrishnan, Vision Research 39 (1999) 131–148, Elsevier Science.
- *Feedback Convolutional Neural Network for Visual Localization and Segmentation*, C. Cao *et al.*, IEEE PAMI vol. 41, 2019.

# Conclusion

In this talk we have presented:

- computer vision tasks of image classification and object detection
- current image benchmark datasets (MNIST, Pascal VOC, ImageNet, MS-COCO)
- the MLP and recent deep learning architectures
- pre-trained, custom trained and used several deep learning architectures
  - MLP and basic CNNs on MNIST
  - VGG16 image classification on ImageNet
  - YOLOv3 object detection on MS-COCO
- Noted new developments (MNIST SOTA, capsules and feedback networks)

## ... But, there are other considerations for computer vision and AI

# Ethics of AI: Other considerations for computer vision and AI

There are many impacts that computer vision and AI already have in society:

- The future of work (impact of CV and AI on jobs, leisure, income)
  - What will the world economy be like when "*AIs*" running on wind and solar power do most of the work?
- Surveillance and privacy (what are appropriate uses of CV and AI)
- User manipulation and monetization
  - e.g., election interference, tracking, advertisement, user behavior
- Laws of robotics (I. Asimov, Robo-Cops, Drones, War machines)

## Ethics of AI Initiatives

- MIT - AI and the Work of the Future (https://workofthefuturecongress.mit.edu/)
- Stanford Institute for Human-Centered Artificial Intelligence (https://hai.stanford.edu/)
- University of Oxford - Future of Humanity Institute (https://www.fhi.ox.ac.uk)
- Electronic Frontier Foundation - Artificial Intelligence & Machine Learning (https://www.eff.org/issues/ai)

# Acknowledgements

We would to thank the following for useful discussions and comments on this work:

- Johann Beukes and Brian Beam, *Levatas*
- K.P. Unnikrishnan, *eNeuroLearn*

*Slides prepared with*
jupyter(https://jupyter.org/)

# References

*Deep Learning with Python*, François Chollet, 2017, Manning Publications, (Chapter 5) https://www.amazon.com/Deep-Learning-Python-Francois-Chollet/dp/1617294438 (https://www.amazon.com/Deep-Learning-Python-Francois-Chollet/dp/1617294438)

*Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, Aurélien Géron, 2017, 1st ed. (second edition in October) https://www.amazon.com/Hands-Machine-Learning-Scikit-Learn-TensorFlow/dp/1491962291 (https://www.amazon.com/Hands-Machine-Learning-Scikit-Learn-TensorFlow/dp/1491962291)

*Deep Learning*, LeCun, Y., Bengio, Y. and Hinton, G. E., Nature, Vol. 521, 2015 (pdf) (http://www.cs.toronto.edu/~hinton/absps/NatureDeepReview.pdf)

### Keras

- https://keras.io (https://keras.io)
- http://github.com/keras-team/keras/ (http://github.com/keras-team/keras/)

# References, cont'd

### VGG16

*Very Deep Convolutional Networks for Large-Scale Image Recognition*, Karen Simonyan & Andrew Zisserman, Visual Geometry Group, Department of Engineering Science, University of Oxford, ICLR 2015. (https://arxiv.org/abs/1409.1556 (https://arxiv.org/abs/1409.1556))

### YOLO

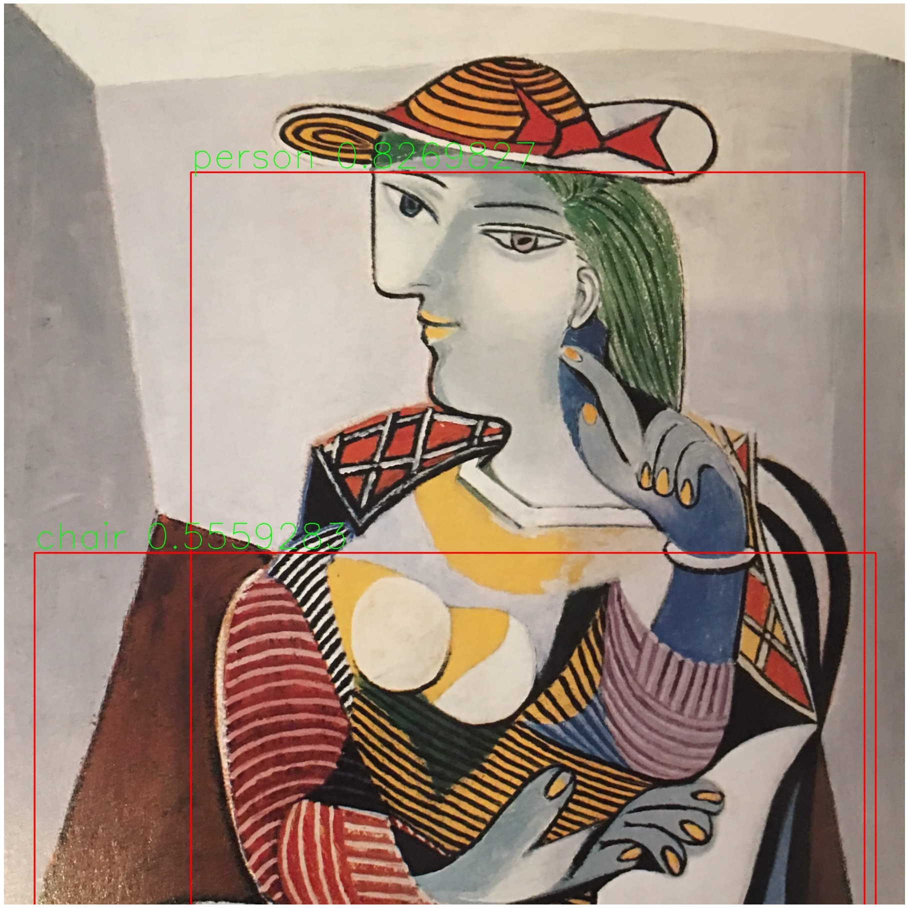
*You Only Look Once: Unified, Real-Time Object Detection*, Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi; University of Washington, Allen Institute for AI, Facebook AI Research, 2015 (https://pjreddie.com/yolo/ (https://pjreddie.com/yolo/))

- https://arxiv.org/abs/1506.2640 (https://arxiv.org/abs/1506.2640) (YOLO)
- https://arxiv.org/abs/1612.08242 (https://arxiv.org/abs/1612.08242) (YOLO-9000, aka YOLOv2)
- https://arxiv.org/abs/1804.02767 (https://arxiv.org/abs/1804.02767) (YOLOv3)

### Computer vision and object detection

*Robust Real-time Object Detection*, Paul Viola and Michael Jones, IJCV 2001 (https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-IJCV-01.pdf)

## Thank you!



Dama Sentada, 1937 (Object detection by YOLOv3)