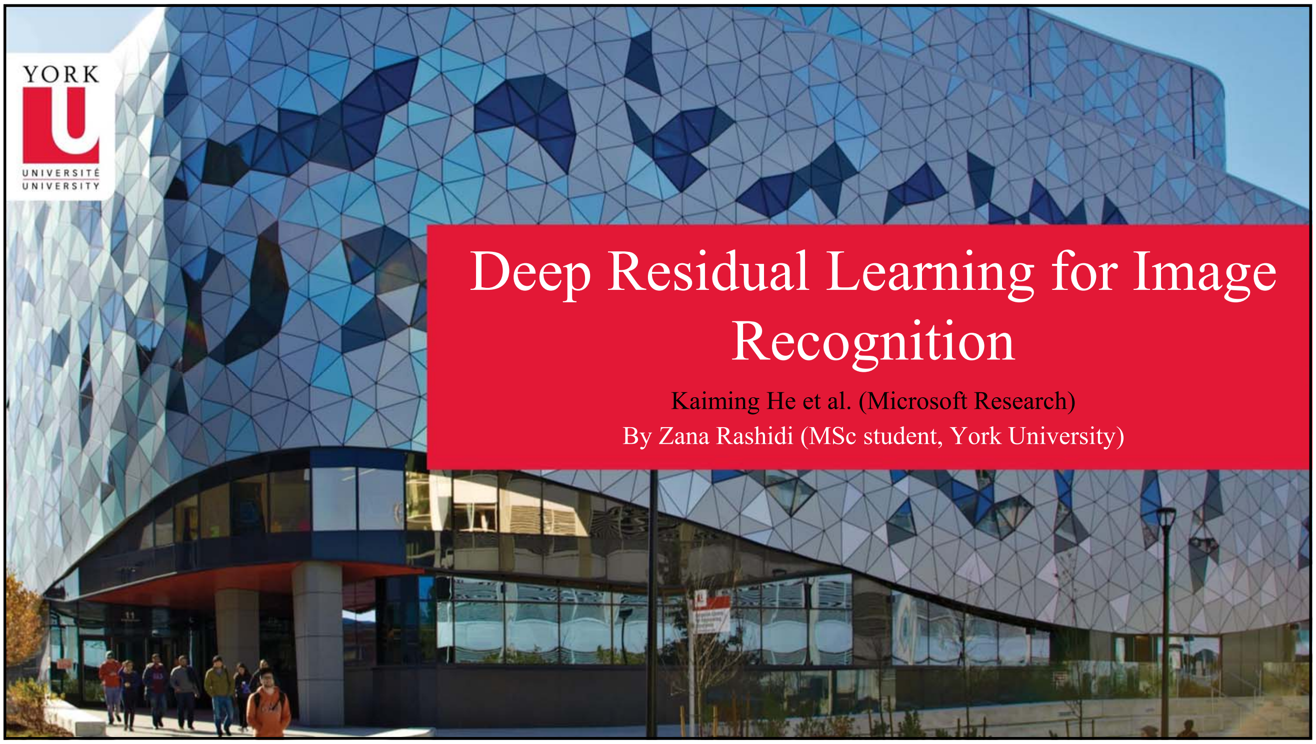


Deep Residual Learning for Image Recognition

Kaiming He et al. (Microsoft Research)
By Zana Rashidi (MSc student, York University)

Introduction

ILSVRC & COCO 2015 Competitions

1st place in **all five main tracks**:

- ImageNet Classification
- ImageNet Detection
- ImageNet Localization
- COCO Detection
- COCO Segmentation

Datasets

ImageNet

- **14,197,122** images
- 27 high-level *categories*
- 21,841 synsets *(subcategories)*
- 1,034,908 images with bounding box annotations

COCO

- **330K** images
- 80 object *categories*
- 1.5M object instances
- 5 captions per image

Tasks

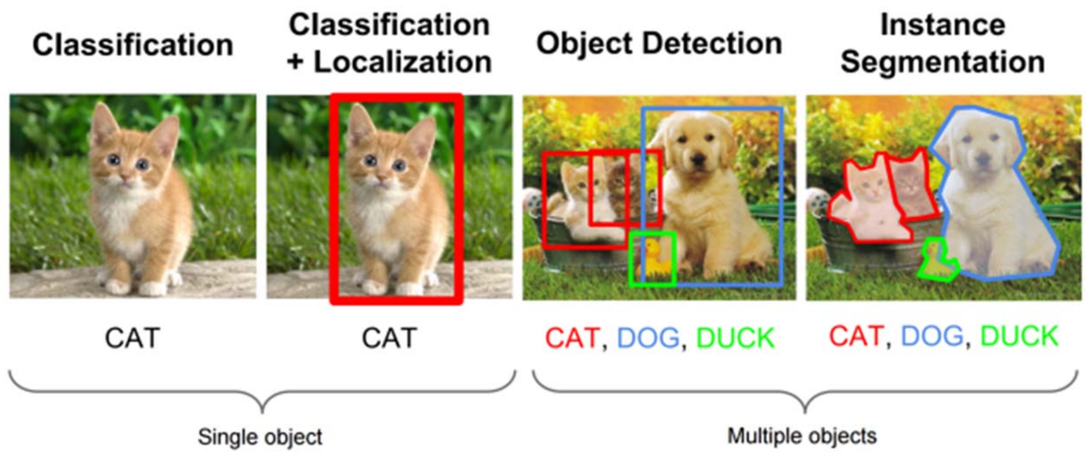


Image from cs231n (Stanford University) Winter 2016

Revolution of Depth

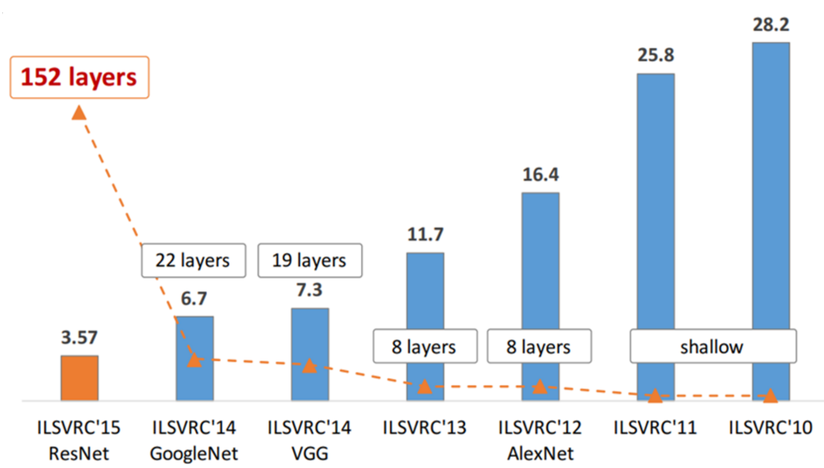
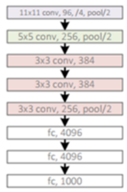


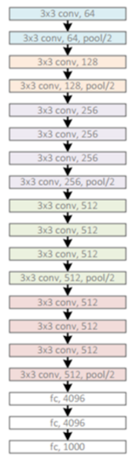
Image from author's slides, ICML 2016

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



GoogleNet, 22 layers
(ILSVRC 2014)

Image from author's slides, ICML 2016

YORK U UNIVERSITÉ UNIVERSITY

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

Image from author's slides, ICML 2016

YORK U UNIVERSITÉ UNIVERSITY

Example

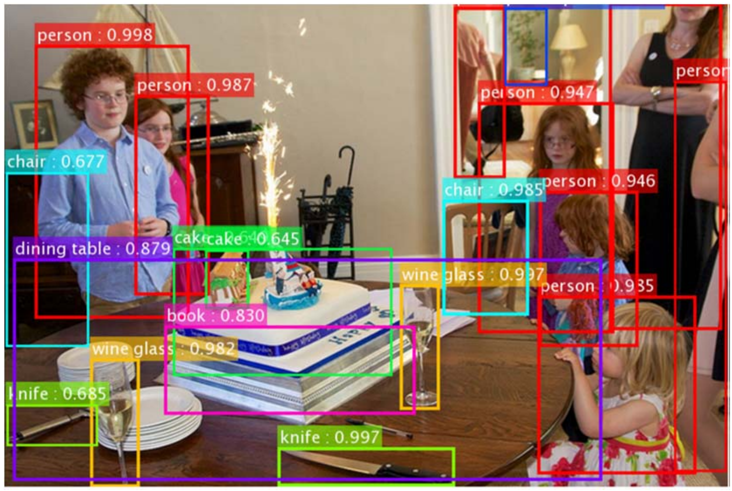


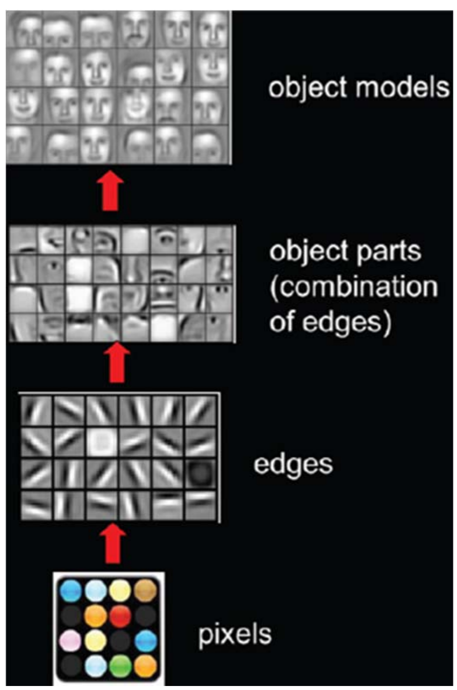
Image from author's slides, ICML 2016

Background

Deep Convolutional Neural Networks

- **Breakthrough** in image classification
- Integrate **low/mid/high-level** features in a *multi-layer* fashion
- Levels of features can be enriched by the number of stacked layers
- Network **depth** is very important

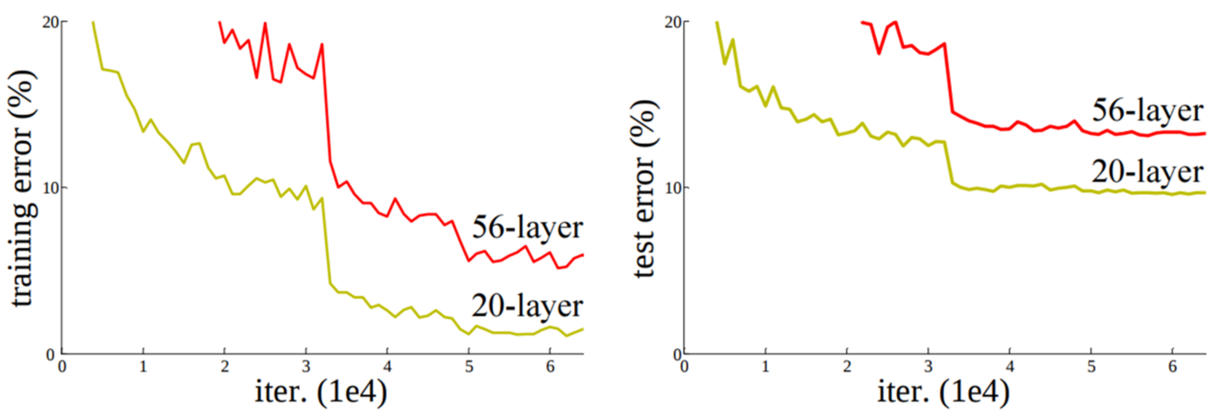
Features (filters)



Deep CNNs

- *Is learning better networks as easy as stacking more layers?*
- **Degradation** problem
  - With ***depth increase***, accuracy gets **saturated**, then degrades rapidly, *not caused by overfitting*, higher training error

Degradation of Deep CNNs



Deep Residual Networks

Address Degradation

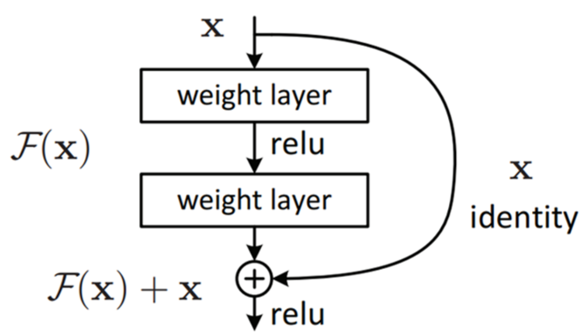
- Consider a shallower architecture and its deeper counterpart
- Solution by *construction*:
  - Add **identity** layers to the shallow learned model to build the deeper model
- The existence of this solution indicates that deeper models should have **no higher training error**, but experiments show:
  - Deeper networks are **unable** to find a solution that is comparable or better than the **constructed** one

Address Degradation (continued)

- So deeper networks are difficult to optimize
- *Deep residual learning* framework
  - Instead of fitting a **few** stacked layers to an underlying mapping
  - Let the layers fit a **residual mapping**
  - Instead of finding the underlying mapping *H(x)*, let the stacked nonlinear layers fit *F(x)=H(x)-x*, so original mapping recasts into *F(x)+x*
- Easier to optimize the residual mapping instead of the original

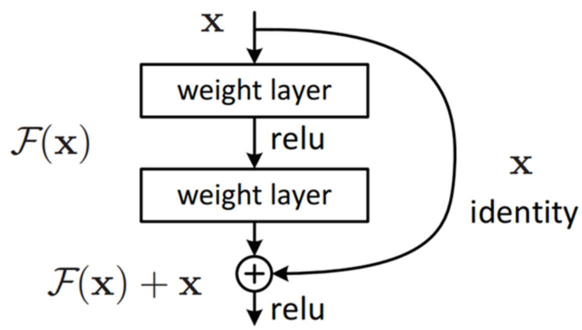
Residual Learning

- If identity mapping was optimal
  - Easier to push **residual** to *zero*
  - Than to **fit** *identity* mapping
- Identity shortcut connections
  - Add to output of stacked layers
  - No extra parameters
  - No computational complexity



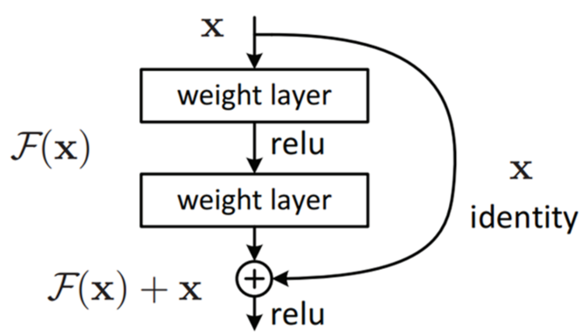
Details

- Adopt residual learning to every few stacked layers
- A building block
  - $y=F(x, W_i)+x$
  - x and y input and output
  - $F(x, W_i)+x$ is the residual mapping to be learned
  - *ReLU* nonlinearity



Details

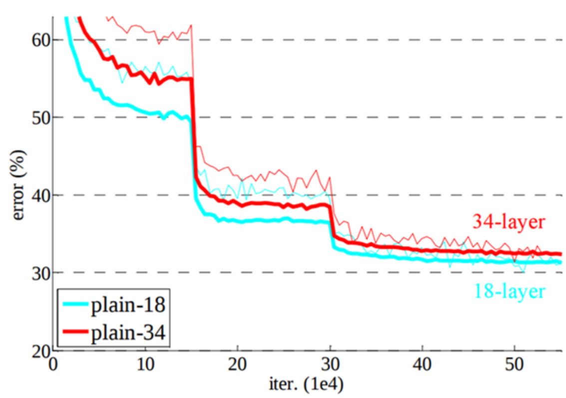
- Dimensions of x and $F(x)$ must be the same
  - Perform *linear* projection
  - $y=F(x,W_i)+W_s x$
  - 2 or 3 layers
  - *Element-wise* addition



Experiments

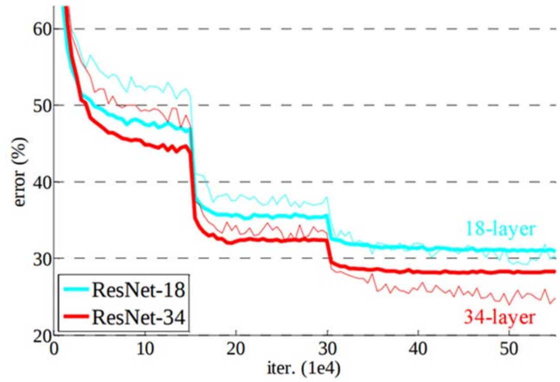
Plain Networks

- 18 and 34 layers
- **Degradation** problem
- 34 layer has **higher** *training* (thin curves) and *validation* (bold curves) error than 18 layer network



Residual Networks

- 18 and 34 layer
- Differ from the plain networks only by *shortcut connections* every two layers
- **Zero-padding** for increasing dimensions
- 34 layer ResNet is **better** than 18 layer ResNet

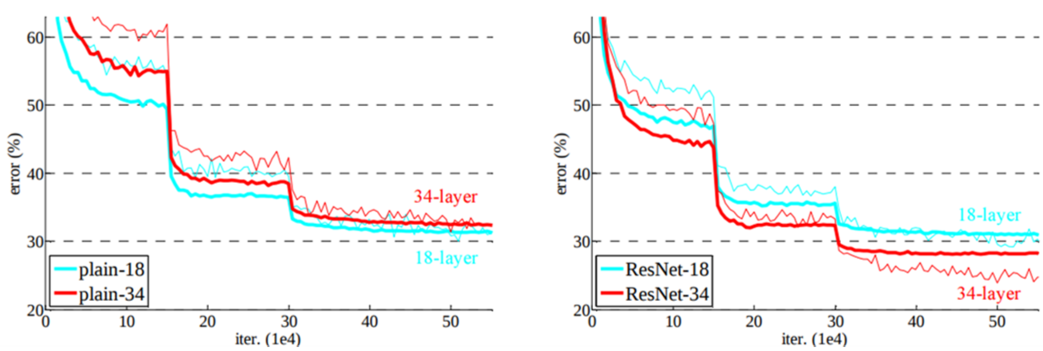


YORK U
UNIVERSITÉ
UNIVERSITY

Comparison

- Reduced *ImageNet* top-1 error by **3.5%**
- Converges *faster*

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**



YORK U
UNIVERSITÉ
UNIVERSITY

Identity *vs.* Projection Shortcuts

- Recall $y=F(x,W_i)+W_s x$

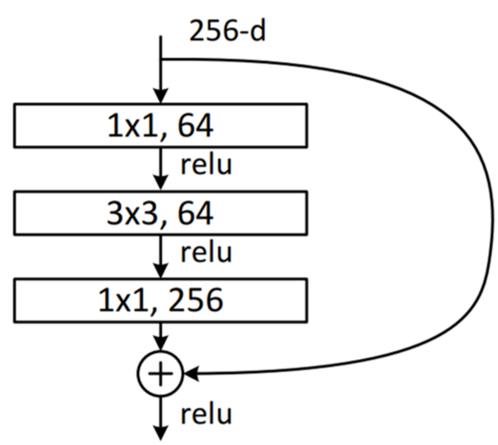
A. Zero-padding for increasing dimension *(parameter free)*
B. Projections for increasing dimension, rest are *identity*
C. All shortcuts are **projections**

model	top-1 err.	top-5 err.
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40

YORK U UNIVERSITÉ UNIVERSITY

Deeper Bottleneck Architecture

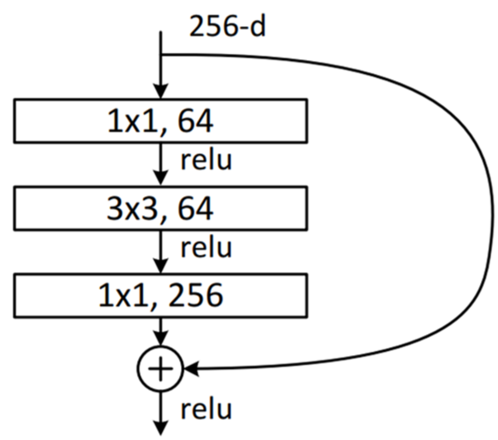
- **Training time** concerns
- Replace residual blocks with 3 layers instead of 2
- 1✕1 convolution for reducing and restoring *dimensions*
- 3✕3 convolution, a **bottleneck** with smaller input/output dimensions



YORK U UNIVERSITÉ UNIVERSITY

50 layer ResNet

- **Replace** each 2 layer residual block with this 3 layer bottleneck block resulting in 50 layers
- Use option **B** for increasing dimensions
- **3.8** billion FLOPs



YORK U UNIVERSITÉ UNIVERSITY

101 layer and 152 layer ResNet

- Add more **bottleneck** blocks
- 152 layer ResNet has **11.3** billion FLOPs
- The deeper, the better
- No *degradation*
- Compared with *state-of-the-art*

model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

YORK U UNIVERSITÉ UNIVERSITY

Results

Object Detection on COCO

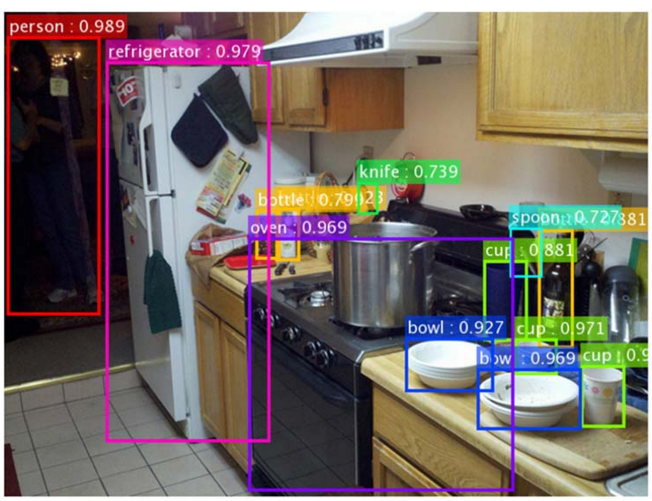


Image from author's slides, ICML 2016

Object Detection on COCO

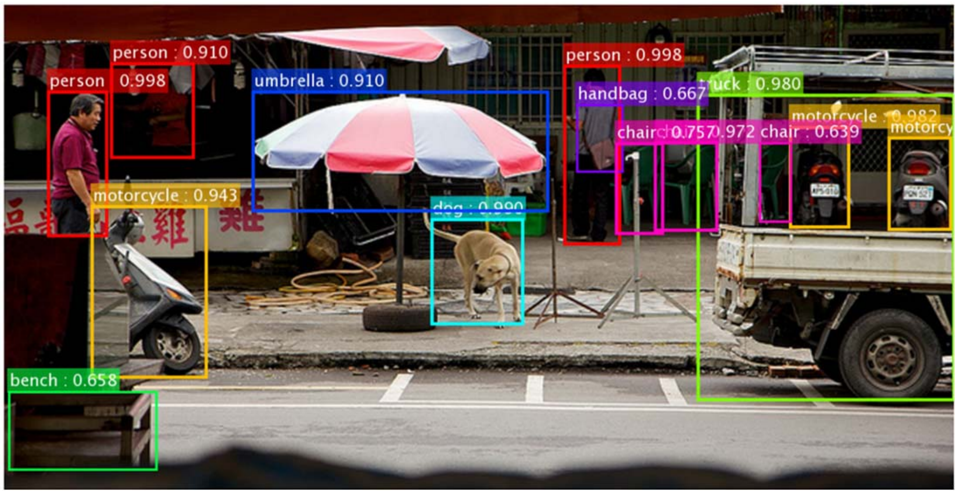
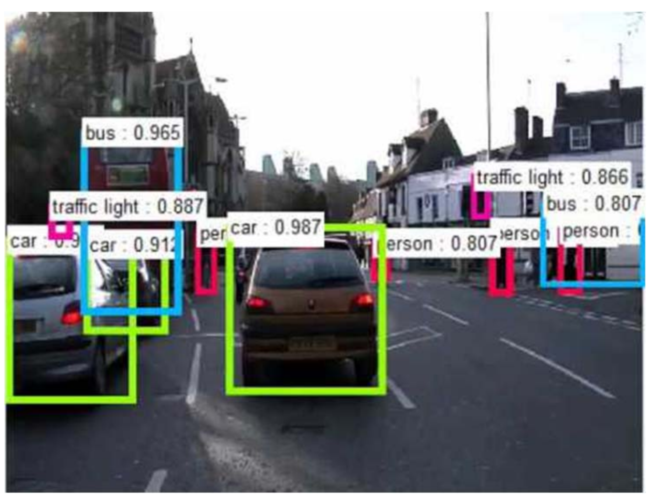


Image from author's slides, ICML 2016

YORK U UNIVERSITÉ UNIVERSITY

Object Detection in the Wild



https://youtu.be/WZmSMkK9VuA

YORK U UNIVERSITÉ UNIVERSITY

Conclusion

Conclusion

- Deep residual learning
  - Ultra deep networks could be **easy** to train
  - Ultra deep networks can gain *accuracy* from **depth**

Applications of ResNet

- Visual Recognition
- Image Generation
- Natural Language Processing
- Speech Recognition
- Advertising
- User Prediction

Resources

- Code written in **Caffe** available in *github*
- Third party implementations in other frameworks
  - Torch
  - Tensorflow
  - Lasagne
  - ...

Thank you!