Detekcja obiektów w czasie rzeczywistym na podstawie małej liczby przykładów uczących

Mateusz Zaborski
M.Zaborski@mini.pw.edu.pl

Faculty of Mathematics and Information Science
Warsaw University of Technology

24.04.2019

Table of Contest

Motivation

Motivation

- Promising development of Computer Vision
- Well-researched image detection problem
- Well-researched image classification problem
- Hot topic in real-time detection
  - Performance
  - Noisy and "zero-shot detection" data
  - Self-driving cars - extra cases
- Good implementaions available

Real-time object detection

R-CNN (CVPR, 2014) [1]

Rich feature hierarchies for accurate object detection and semantic segmentation

Ross Girshick[1] Jeff Donahue[1,2] Trevor Darrell[1,2] Jitendra Malik[1]
[1]UC Berkeley and [2]ICSI
{rbg,jdonahue,trevor,malik}@eecs.berkeley.edu

Figure: R-CNN paper

R-CNN

Short history

- HOG and SIFT had been used for 10 years (for object detection and recogniton)
  - Gradients, histograms, keypoints
- CNNs saw heavy use in the 1990s
- In 2012, Krizhevsky et al. [2] rekindled interest in CNNs by showing substantially higher image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

R-CNN

Overview

- Try CNN classification on the PASCAL VOC Challenge
- Object detection performance has plateaued in the last few years
- Proposition a simple and scalable detection algorithm (that improves)
- Bridging the chasm between image classification and object detection
- Supervised pretraining on a large auxiliary dataset (ILSVRC), followed by domain-specific fine-tuning on a small dataset (PASCAL), is an effective paradigm for learning high-capacity CNNs when data is scarce
- Region proposals + CNN = R-CNN

R-CNN

Object detection

System consists three modules:

1. Generates category-independent region proposals - Selective Search (can be other)
2. Large convolutional neural network that extracts a fixed-length feature vector from each region
3. Set of classspecific linear SVMs

R-CNN

Object detection

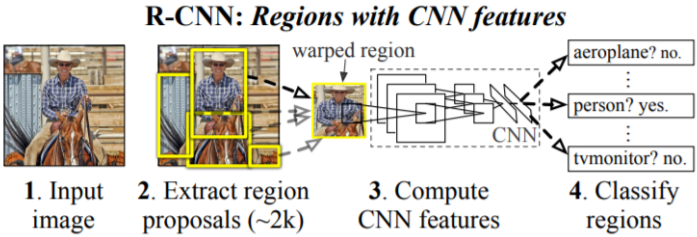


Figure: R-CNN principal

R-CNN

Selective search

Figure: Selective Search idea

http://vision.stanford.edu/teaching/cs231b_spring1415/slides/ssearch_schuyler.pdf

R-CNN

Feature extraction

- Extract a 4096-dimensional feature vector
- Using the Caffe implementation of the CNN described by Krizhevsky et al.
  - AlexNet
- Inputs of a fixed 227 x 227 pixel size

R-CNN

Feature extraction

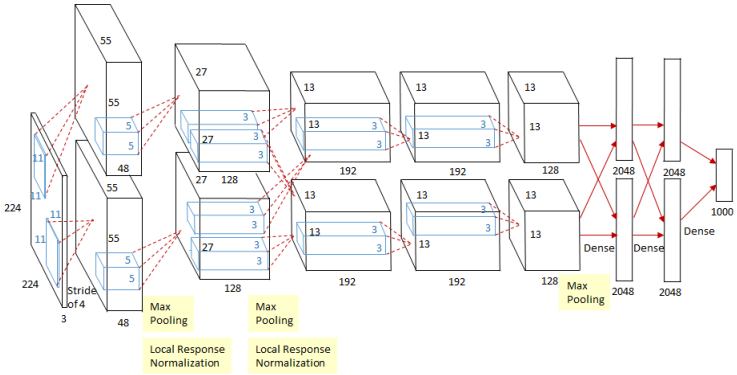


Figure: AlexNet

R-CNN

Test-time detection

- Extract around 2000 region proposals with Selective Search
- Forward propagate it through the CNN in order to read off features
- Score each extracted feature vector using the SVM trained for that class (for each class)
- Rejects a region if it has an intersection-over-union (IoU) overlap with a higher scoring selected region larger than a learned threshold
- 13s/image on a GPU or 53s/image on a CPU)

R-CNN

Training

- Pre-trained the CNN on a large auxiliary dataset (ILSVRC 2012) with image-level annotations (no bounding box labels)
- Domain-specific fine-tuning
  - Warped region proposals from VOC
  - Replacing the CNN's ImageNet-specific 1000-way classification layer with a randomly initialized 21-way classification layer
  - All region proposals with >= 0.5 IoU overlap with a ground-truth box as positives
  - SGD at a learning rate of 0.001 (1/10th of the initial pre-training rate)
  - Change IoU overlap threshold using grid search (to optimize validation set result)
  - Optimizeing one linear SVM per class

R-CNN

Bounding Box Regression

Train a linear regression model to predict a new detection window given the features for a selective search region proposal

Figure: R-CNN Bounding Box Regression

R-CNN

Results on PASCAL VOC 2010-12

- Validated all design decisions and hyperparameters on the VOC 2007 dataset
- Fine-tuned the CNN on VOC 2012 train and optimized our detection SVMs on VOC 2012 trainval
- Extra Bounding Box regression model compared

R-CNN

Results on PASCAL VOC 2010-12

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [17]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [32]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [35]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [15]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

Table 1: Detection average precision (%) on VOC 2010 test. R-CNN is most directly comparable to UVA and Regionlets since all methods use selective search region proposals. Bounding box regression (BB) is described in Section 3.4. At publication time, SegDPM was the top-performer on the PASCAL VOC leaderboard. †DPM and SegDPM use context rescoring not used by the other methods.

Figure: PASCAL VOC 2010 results

Similar performance (53.3% mAP) on VOC 2011/12 test

Fast R-CNN (ICCV, 2015) [3]

Figure: Fast R-CNN paper

Fast R-CNN

Overview

- Trains the VGG-16
- Faster training
- Much faster object detection
- One-stage pipeline
- Higher mAP on PASCAL VOC 2012

Fast R-CNN

Principle[3]

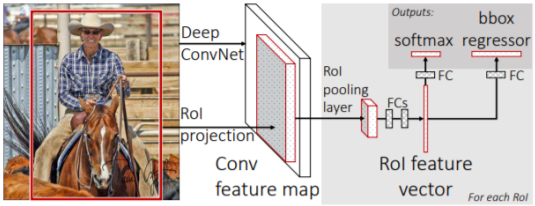


Figure 1. Fast R-CNN architecture. An input image and multiple regions of interest (RoIs) are input into a fully convolutional network. Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers (FCs). The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets. The architecture is trained end-to-end with a multi-task loss.

Fast R-CNN

Multi-task loss

Multi-task loss L on each labeled RoI used to jointly train for classification and bounding-box regression.

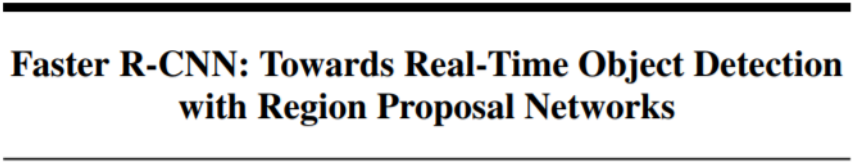
$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1]L_{\text{loc}}(t^u, v)$$

$$L_{\text{cls}}(p, u) = -\log p_u$$

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t_i^u - v_i)$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

Faster R-CNN (NIPS, 2015) [4]



Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Shaoqing Ren* **Kaiming He** **Ross Girshick** **Jian Sun**
Microsoft Research
{v-shren, kahe, rbg, jiansun}@microsoft.com

Figure: Faster R-CNN paper

Faster R-CNN

Overview

- Region Proposal Network (RPN) introduced
  - Entire image as input
  - Shares full-image convolutional features with the detection network
- RPN - fully-convolutional network
  - Simultaneously predicts object bounds and objectness scores at each position
- Region proposal step is nearly cost-free
- Frame rate of 5fps (including all steps) on a GPU
  - Achieving state-of-the-art object detection accuracy on PASCAL VOC 2007 (73.2% mAP)

Faster R-CNN

RPN anchors

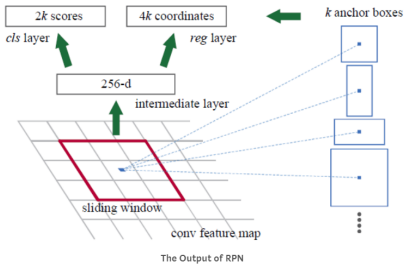


Figure: Anchors in RPN

https://towardsdatascience.com/review-faster-r-cnn-object-detection-f5685cb30202

YOLO (CVPR, 2016) [5]

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¶
http://pjreddie.com/yolo/

Figure: YOLO paper

YOLO

Overview

- Processes images in real-time at 45 frames per second
  - Smaller version of the network (Fast YOLO) - 155 frames per second
- Object detection as a regression problem to spatially separated bounding boxes and associated class probabilities
- Makes more localization errors but is less likely to predict false positives on background

YOLO [5]

Model

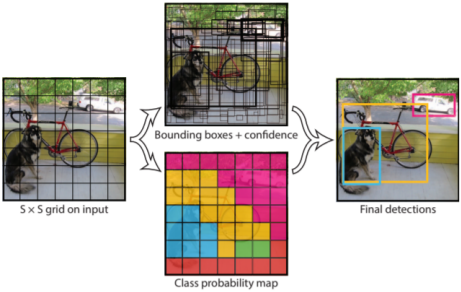


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

YOLO

Principal

- Dividing entire input image into an SxS grid
- Each grid cell predicts only one object
  - B boundary boxes for each cell
    - Each bounding box consists of 5 predictions: x, y, w, h and confidence
  - Each grid cell also predicts C conditional class probabilities (only one set)

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

YOLO

YOLO architecture [5]

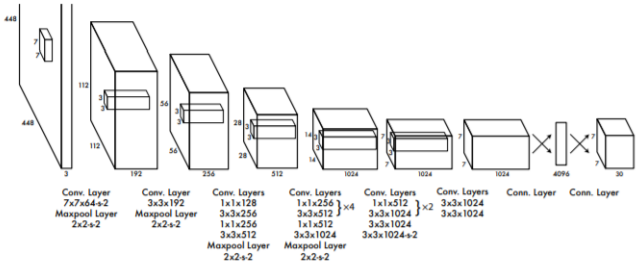
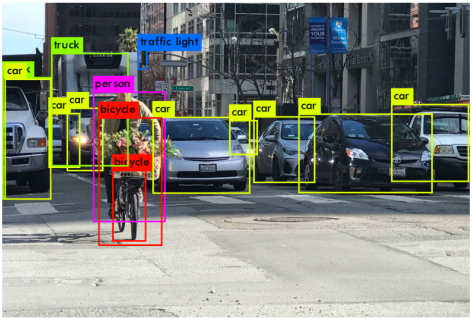


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

YOLO

YOLO real-time detection



https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

SSD (ECCV, 2016) [6]

SSD: Single Shot MultiBox Detector

Wei Liu[1], Dragomir Anguelov[2], Dumitru Erhan[3], Christian Szegedy[3],
Scott Reed[4], Cheng-Yang Fu[1], Alexander C. Berg[1]

[1]UNC Chapel Hill [2]Zoox Inc. [3]Google Inc. [4]University of Michigan, Ann-Arbor
[1]wliu@cs.unc.edu, [2]drago@zoox.com, [3]{dumitru,szegedy}@google.com,
[4]reedscot@umich.edu, [1]{cyfu,aberg}@cs.unc.edu

Figure: SSD paper

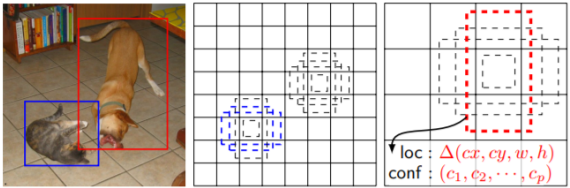
SSD

Overview

- Faster than YOLO
- Beats the accuracy of the Faster R-CNN (as mAP measure)
- No need of the region proposal network
- Two main parts
  - Extraction of features
  - Convolution filters to detect objects
- End-to-end training

SSD

Principal [6]

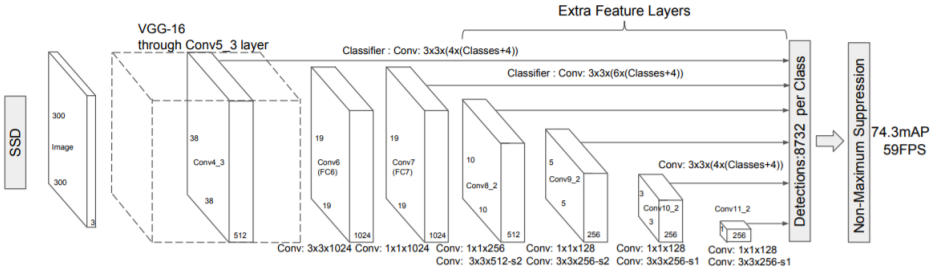


(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

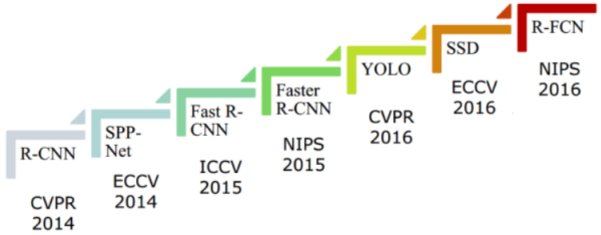
Fig. 1: **SSD framework.** (a) SSD only needs an input image and ground truth boxes for each object during training. In a convolutional fashion, we evaluate a small set (e.g. 4) of default boxes of different aspect ratios at each location in several feature maps with different scales (e.g. 8×8 and 4×4 in (b) and (c)). For each default box, we predict both the shape offsets and the confidences for all object categories ($(c_1, c_2, \cdots, c_p)$). At training time, we first match these default boxes to the ground truth boxes. For example, we have matched two default boxes with the cat and one with the dog, which are treated as positives and the rest as negatives. The model loss is a weighted sum between localization loss (e.g. Smooth L1 [6]) and confidence loss (e.g. Softmax).

SSD

Architecture [6]



Consequent papers



https://towardsdatascience.com/review-r-fcn-positive-sensitive-score-maps-object-detection-91cd2389345c

Consequent papers

- Fast/Faster R-CNN
  - R-FCN [9]
- YOLO
  - YOLO9000: Better, Faster, Stronger [10]
  - YOLOv3: An Incremental Improvement [11]

Limited data training

Problem overview

- Training set is often limited
- Keep end-to-end training
  - Object detection
  - Object classification

Few-shot Object Detection via Feature Reweighting (2018) [13]

Few-shot Object Detection via Feature Reweighting

Bingyi Kang[1*], Zhuang Liu[2*], Xin Wang[2], Fisher Yu[2], Jiashi Feng[1], Trevor Darrell[2]
[1]National University of Singapore [2]University of California, Berkeley

Figure: Few-shot Object Detection via Feature Reweighting paper

Few-shot Object Detection via Feature Reweighting

Overview

- Few-shot technique into object detection
- Two models
  - Base model
  - Meta-model
- Based on YOLO v2
- DarkNet-19 as feature extractor
- Weights added to features

Few-shot Object Detection via Feature Reweighting

Model[13]

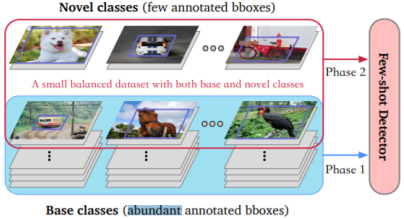


Figure 1: We propose a novel few-shot detection model that trains a basic detection model and a meta-model on base classes with sufficiently many samples in the first phase. In the second phase the meta-model is trained to fast adapt the detection model to detect novel objects by reweighing its intermediate features to be more discriminative and sensitive to the novel classes with a few examples.

Few-shot Object Detection via Feature Reweighting

Training

- Base training
  - Only with base classes
  - Each iteraton N images used for reweighting - producing N vectors
- Few-shot fine-tuning
  - Training the model on both base and novel classes
  - Base and novel set has only k-shots

Few-shot Object Detection via Feature Reweighting

Architecture[13]

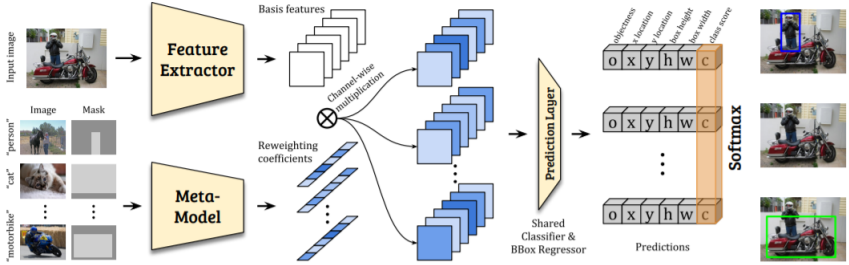


Figure 2: The architecture of our proposed few-shot detection model. It consists of a base feature extractor and a meta-model. The base model follows the one-stage detector architecture and directly regresses the objectness score (o), bounding box location (x, y, h, w) and classification score (c). The Metanet is trained to map N classes of inputs to N feature reweighting coefficients, each responsible for adjusting the basis features from the Darknet to detect the objects from the corresponding class. A softmax based classification score normalization is imposed on the final output.

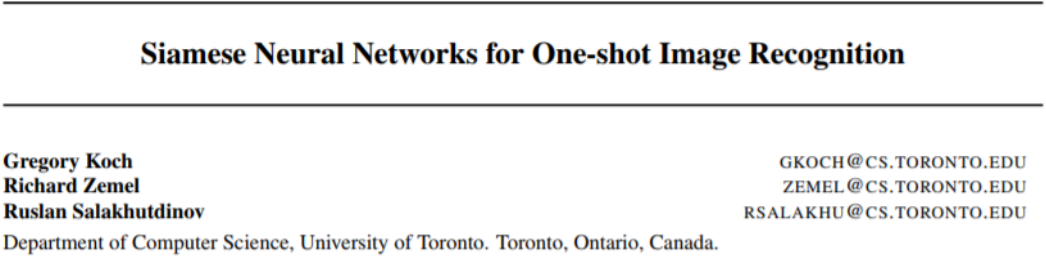
Few-shot Object Detection via Feature Reweighting

Results [13]

	Novel Set 1					Novel Set 2					Novel Set 3				
Method / Shot	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
YOLO-joint	0.0	0.0	1.8	1.8	1.8	0	0.1	0	1.8	0	1.8	1.8	1.8	3.6	3.9
YOLO-ft	3.2	6.5	6.4	7.5	12.3	8.2	3.8	3.5	3.5	7.8	8.1	7.4	7.6	9.5	10.5
YOLO-ft-full	6.6	10.7	12.5	24.8	38.6	12.5	4.2	11.6	16.1	33.9	13.0	15.9	15.0	32.2	38.4
Ours	**14.8**	**15.5**	**26.7**	**33.9**	**47.2**	**15.7**	**15.3**	**22.7**	**30.1**	**39.2**	**19.2**	**21.7**	**25.7**	**40.6**	**41.3**

Table 2: Few-shot detection performance (mAP) on the PASCAL VOC dataset. We evaluate the performance on three different sets of novel categories. Our model consistently outperforms baseline methods.

Siamese Neural Networks for One-shot Image Recognition (ICML, 2015) [7]



Siamese Neural Networks for One-shot Image Recognition

Gregory Koch GKOCH@CS.TORONTO.EDU
Richard Zemel ZEMEL@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Department of Computer Science, University of Toronto. Toronto, Ontario, Canada.

Figure: Siamese Neural Network paper

Siamese Neural Networks for One-shot Image Recognition

Omniglot dataset [7]

Figure 1. Example of a 20-way one-shot classification task using the Omniglot dataset. The lone test image is shown above the grid of 20 images representing the possible unseen classes that we can choose for the test image. These 20 images are our only known examples of each of those classes.

Siamese Neural Networks for One-shot Image Recognition[7]

Overview

- One-shot learning
- Twin networks with distinct inputs
  - Identical weights
- Energy function at the top

Siamese Neural Networks for One-shot Image Recognition

Architecture[7]

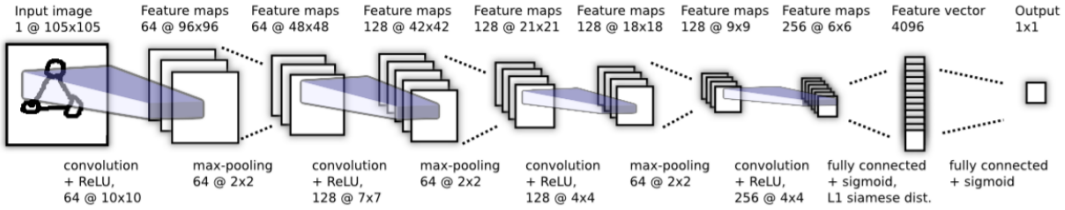


Figure 4. Best convolutional architecture selected for verification task. Siamese twin is not depicted, but joins immediately after the 4096 unit fully-connected layer where the L1 component-wise distance between vectors is computed.

Siamese Neural Networks for One-shot Image Recognition

Learning[7]

$$\mathcal{L}(x_1^{(i)}, x_2^{(i)}) = \mathbf{y}(x_1^{(i)}, x_2^{(i)}) \log \mathbf{p}(x_1^{(i)}, x_2^{(i)}) + (1 - \mathbf{y}(x_1^{(i)}, x_2^{(i)})) \log (1 - \mathbf{p}(x_1^{(i)}, x_2^{(i)})) + \boldsymbol{\lambda}^T |\mathbf{w}|^2$$

Figure: Loss function

- Backpropagation algorithm
- All network weights in the convolutional layers initialized from a normal distribution
- Annealing the learning rate - 1% per epoch
- Stop of the model at the best epoch according to the one-shot validation error
- Training set augmentation with small affine distortions

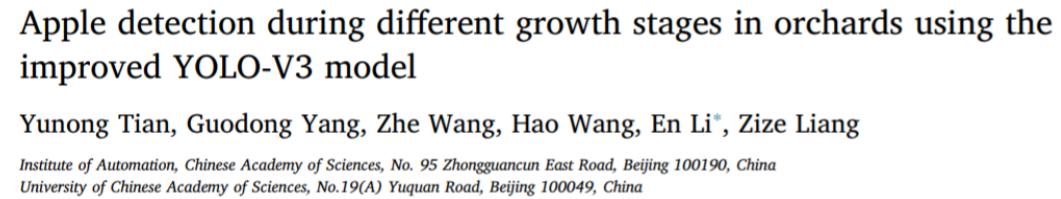
Siamese Neural Networks for One-shot Image Recognition

Results[7]

Table 2. Comparing best one-shot accuracy from each type of network against baselines.

Method	Test
Humans	95.5
Hierarchical Bayesian Program Learning	95.2
Affine model	81.8
Hierarchical Deep	65.2
Deep Boltzmann Machine	62.0
Simple Stroke	35.2
1-Nearest Neighbor	21.7
Siamese Neural Net	58.3
Convolutional Siamese Net	92.0

Apple detection during different growth stages in orchards using the improved YOLO-V3 model (2019) [12]



Apple detection during different growth stages in orchards using the improved YOLO-V3 model

Yunong Tian, Guodong Yang, Zhe Wang, Hao Wang, En Li*, Zize Liang

Institute of Automation, Chinese Academy of Sciences, No. 95 Zhongguancun East Road, Beijing 100190, China
University of Chinese Academy of Sciences, No.19(A) Yuquan Road, Beijing 100049, China

Figure: Apple detection during different growth stages in orchards using the improved YOLO-V3 model paper

Apple detection during different growth stages in orchards using the improved YOLO-V3 model

Overview

- Detect apples during a different growth stage
- Improved YOLO-V3 model (DenseNet)
- 960 images taken
- Images for training set strongly augmented (10x)

Apple detection during different growth stages in orchards using the improved YOLO-V3 model

Apple detection [12]

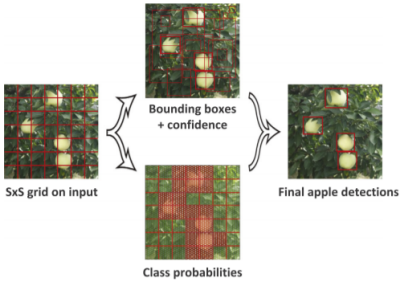


Fig. 2. YOLO Detection.

Apple detection during different growth stages in orchards using the improved YOLO-V3 model

Results [12]

Table 5
F1 Scores, IoU and average detection time for several models.

Models	YOLO-V2	YOLO-V3	Faster R-CNN with VGG16 net	YOLOV3-dense
F1 score	0.738	0.793	0.801	0.817
IoU	0.805	0.869	0.873	0.896
Average time (s)	0.273	0.296	2.42	0.304

Conclusion

Conclusion

- Object detection algorithms have evolved over the last few years
- Deep learning is the key approach
- Training using limited set is new issue

Bibliography I

[1] R. Girshicket et al. „Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR, 2014

[2] A. Krizhevsky et al. „ImageNet classification with deep convolutional neural networks", NIPS, 2012

[3] R. Girshick and Microsoft Research „Fast R-CNN", ICCV, 2015

[4] S. Ren et al. „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS, 2015

[5] R. Redmon et al. „You Only Look Once: Unified, Real-Time Object Detection", CVPR, 2016

[6] W. Liu et al. „SSD: Single Shot MultiBox Detector", ECCV, 2016

Bibliography II

[7] G. Koch et al. „Siamese neural networks for one-shot image recognition", ICML, 2015

[8] Anonymous authors, „A Closer Look at Few-Shot Classification", ICLR, 2019 (under review)

[9] J. Dai et al., „R-FCN: Object Detection via Region-based Fully Convolutional Networks", NIPS, 2016

[10] Redmon, Joseph, and Ali Farhadi, „YOLO9000: better, faster, stronger", CVPR, 2017

[11] Redmon, Joseph, and Ali Farhadi, „YOLOv3: An Incremental Improvement", Tech report, 2018

Bibliography III

[12] Y. Tian et al., „Apple detection during different growth stages in orchards using the improved YOLO-V3 model", Computers and Electronics in Agriculture, 2019

[13] B. Kang et al., „Few-shot Object Detection via Feature Reweighting", preprint, 2018