Advanced Models for Vision

EE807: Recent Advances in Deep Learning

Lecture 18

Slide made by

Hyungwon Choi and Hankook Lee

KAIST EE

Korea Advanced Institute of Science and Technology

Table of Contents

Table of Contents

What is Object Detection?

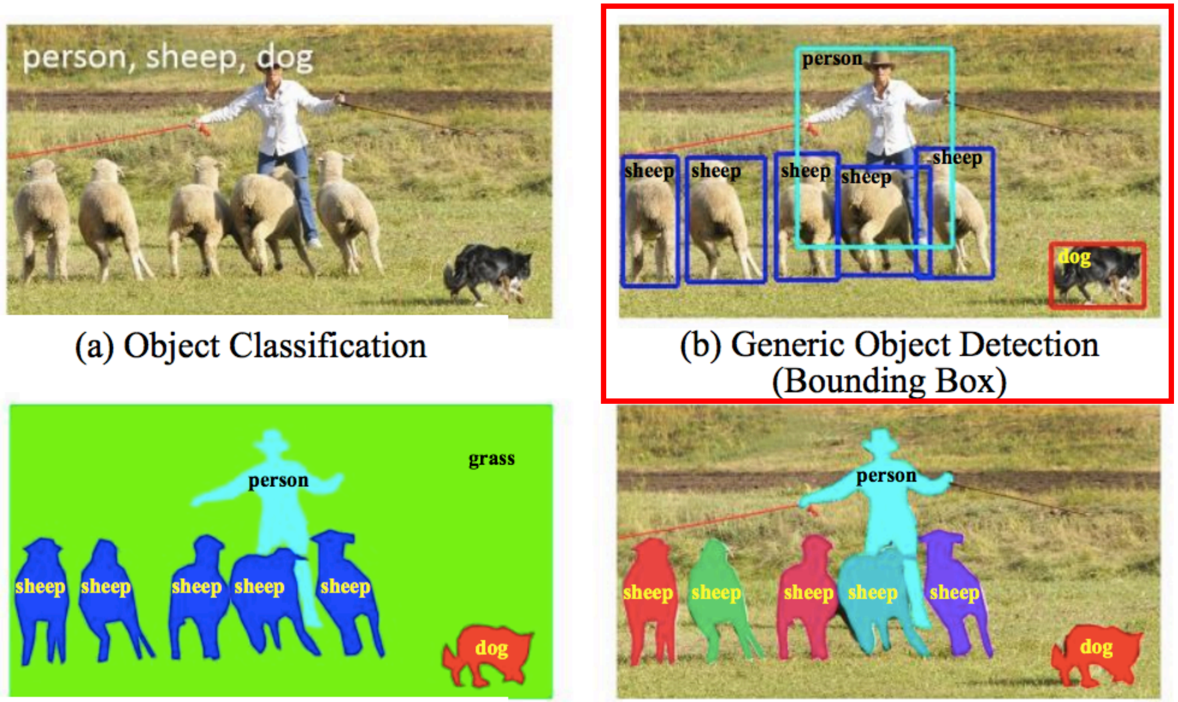
- Goal: **Predict** both **concepts(class)** and **locations** of **every object** in a scene
  - Classification + bounding-box regression (coordinates)
  - More complicated than single object classification

“Image level”

“Bounding box level”

“Pixel level”

“Instance level”



(a) Object Classification

(b) Generic Object Detection (Bounding Box)

(c) Semantic Segmentation

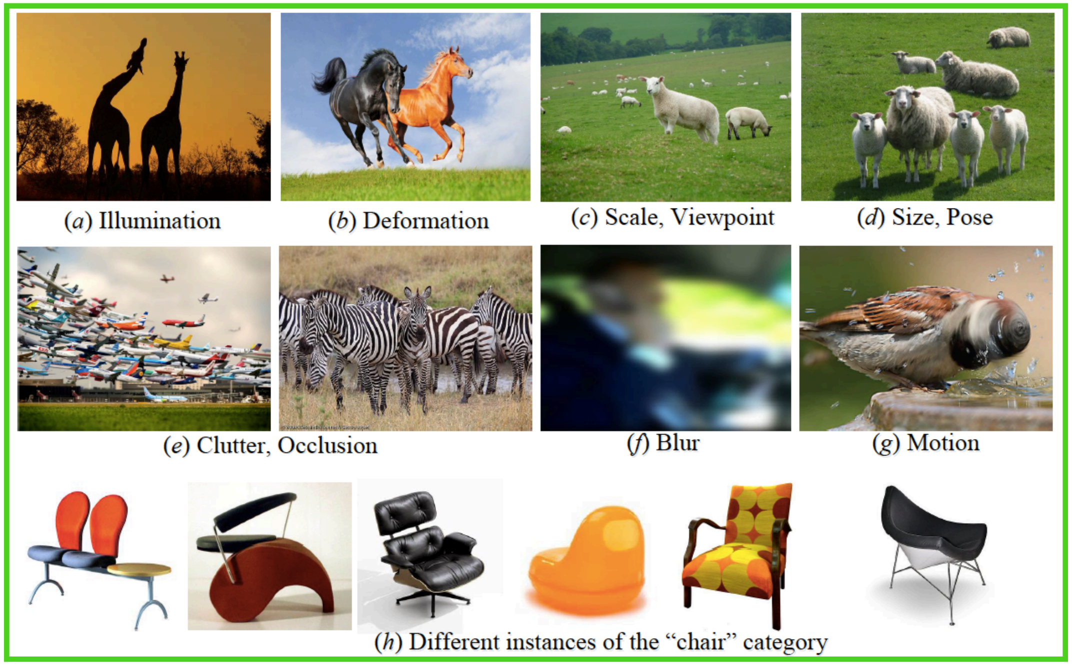
(d) Object Instance Segmetation

- One of the most **fundamental** and **challenging** problems in computer vision

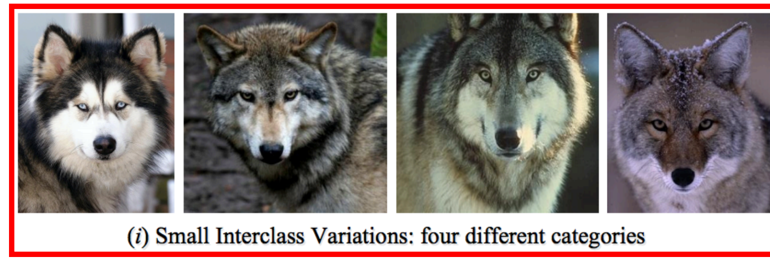
*source: https://arxiv.org/pdf/1809.02165.pdf

Object Detection: Challenges

- **Accuracy**
  - Vast range of intraclass variations **(a-h)**
  - Small interclass variations **(i)**



(*a*) Illumination (*b*) Deformation (*c*) Scale, Viewpoint (*d*) Size, Pose
(*e*) Clutter, Occlusion (*f*) Blur (*g*) Motion
(*h*) Different instances of the "chair" category

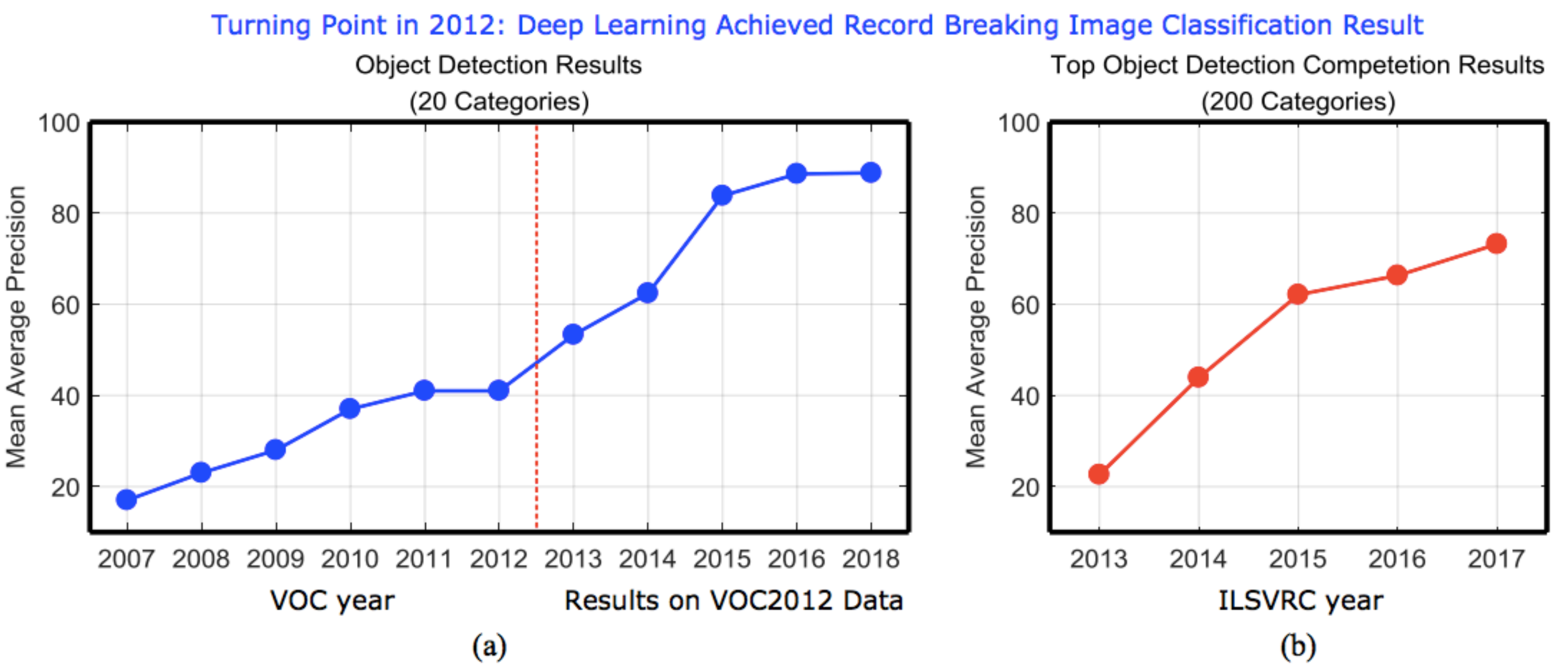


(*i*) Small Interclass Variations: four different categories

- **Efficiency**
  - Need to localize/recognize **all object instances** with different scales
  - Increasing needs on sufficiently **high frame rate** (towards **real-time)**

*source: https://arxiv.org/pdf/1809.02165.pdf

- **Recent evolution** of object detection performance

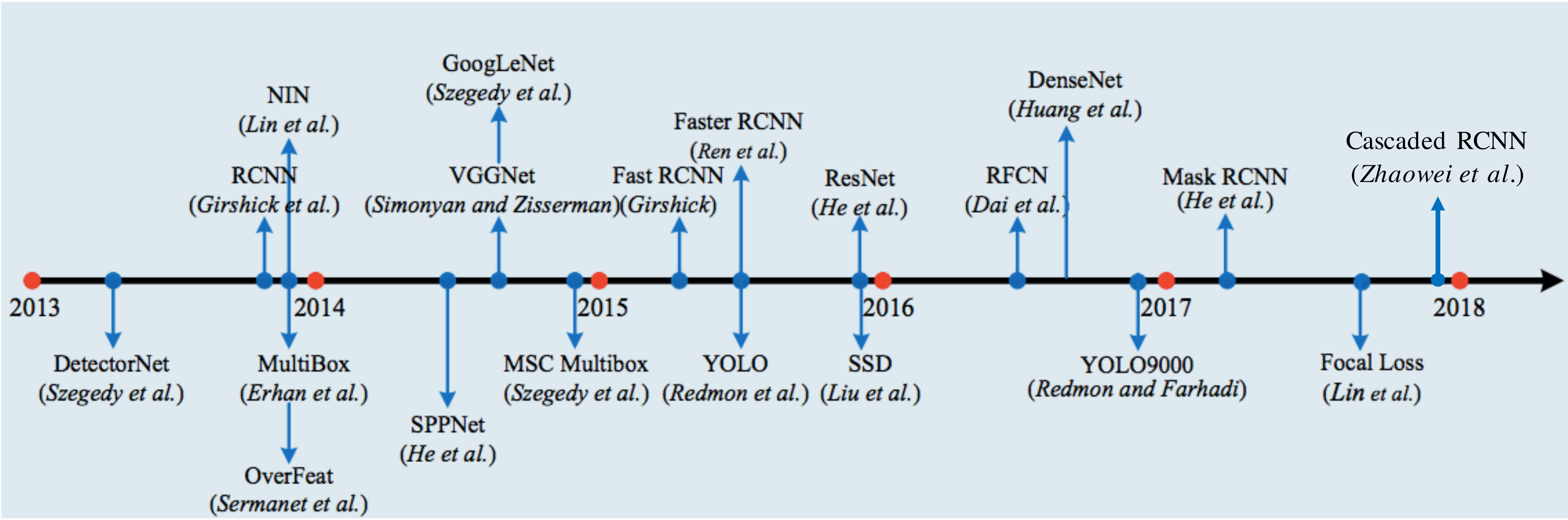


- Generic object detection performance **steadily increased** since 2012
  - Thanks to evolution of deep CNNs
  - Similar tendency with ImageNet classification performance

*source: https://arxiv.org/pdf/1809.02165.pdf

Object Detection: Overview

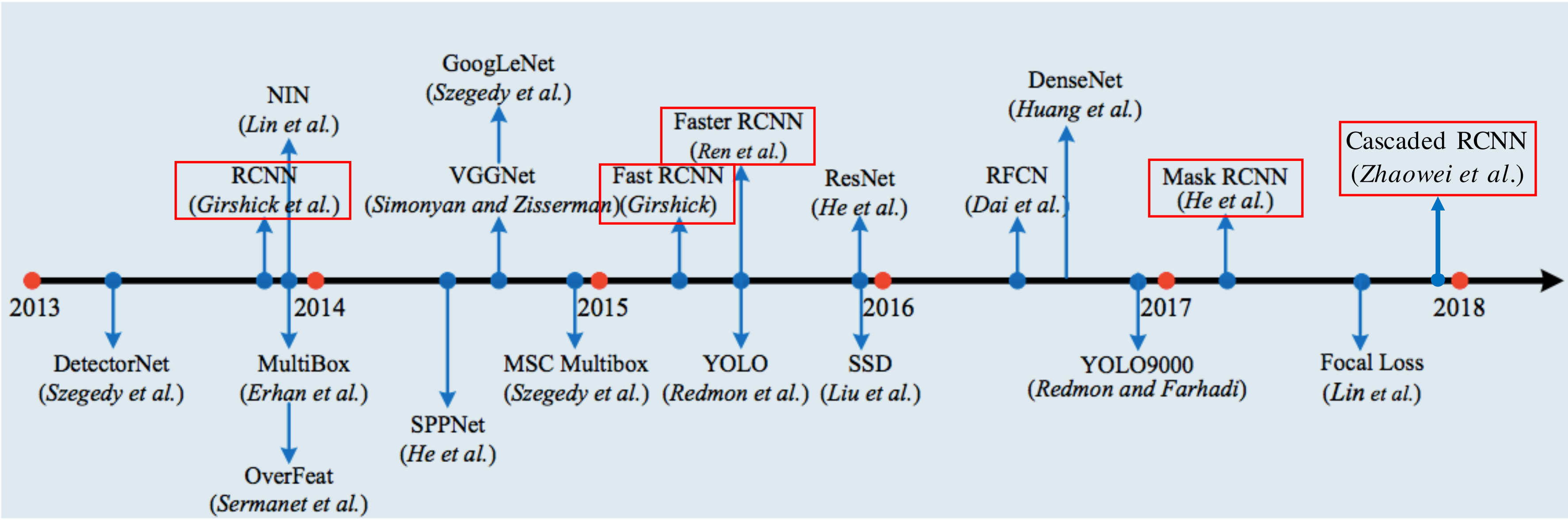
- **Milestones** in **object detection** based on the time of their **first arXiv** version



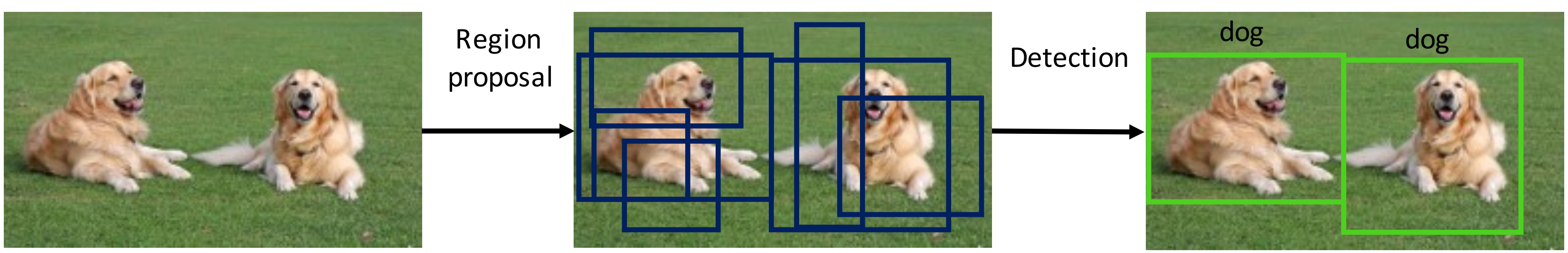
*source: https://arxiv.org/pdf/1809.02165.pdf

Object Detection: Overview

- **Milestones** in **object detection** based on the time of their **first arXiv** version

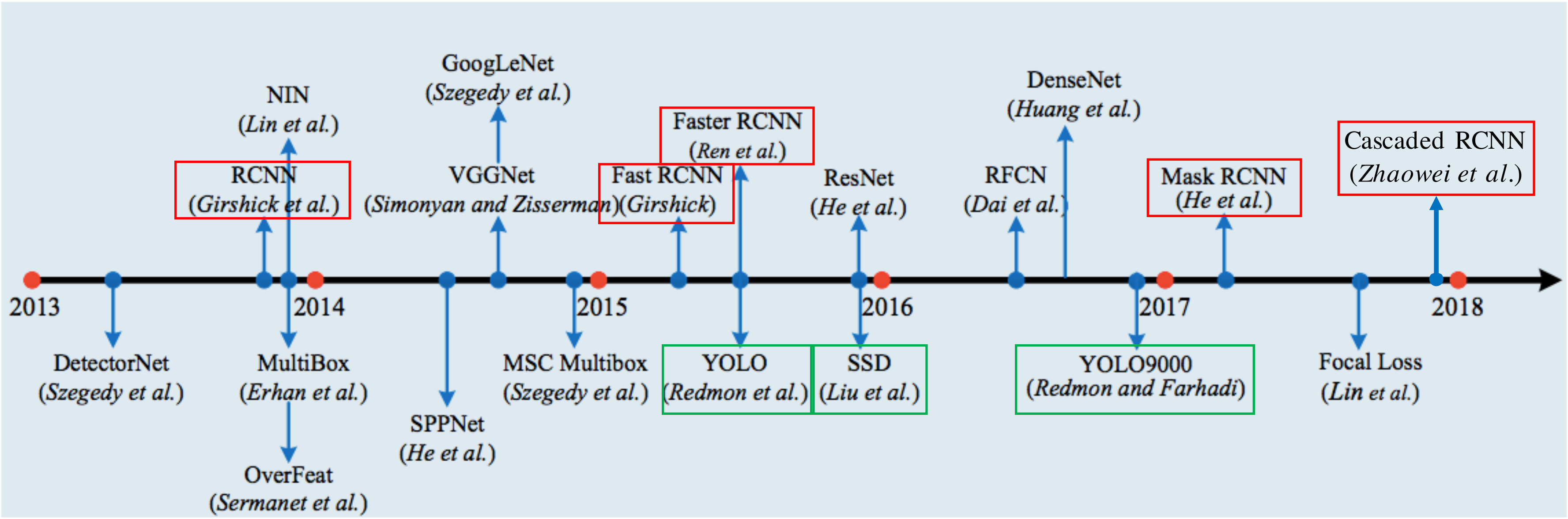


- Region-based detectors
  - **Two-stage** framework
  - Region proposals → Detection (bbox regression + classification)

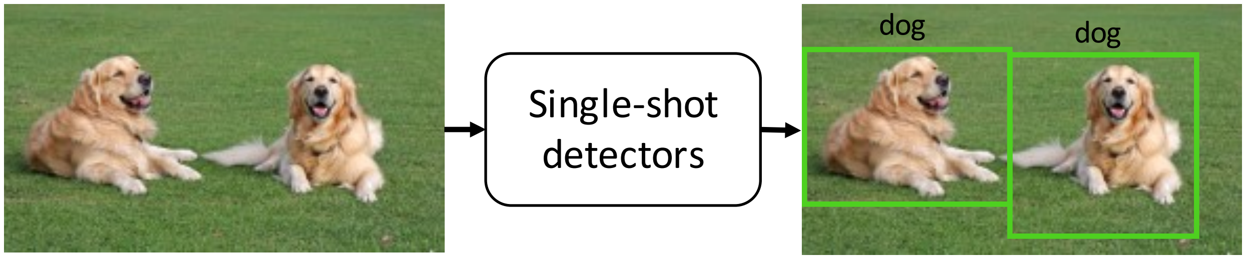


Object Detection: Overview

- **Milestones** in **object detection** based on the time of their **first arXiv** version



- Single-shot detectors
  - **Region-proposal-free** methods
  - Unified, single-stage framework



Next: Region-based Detectors

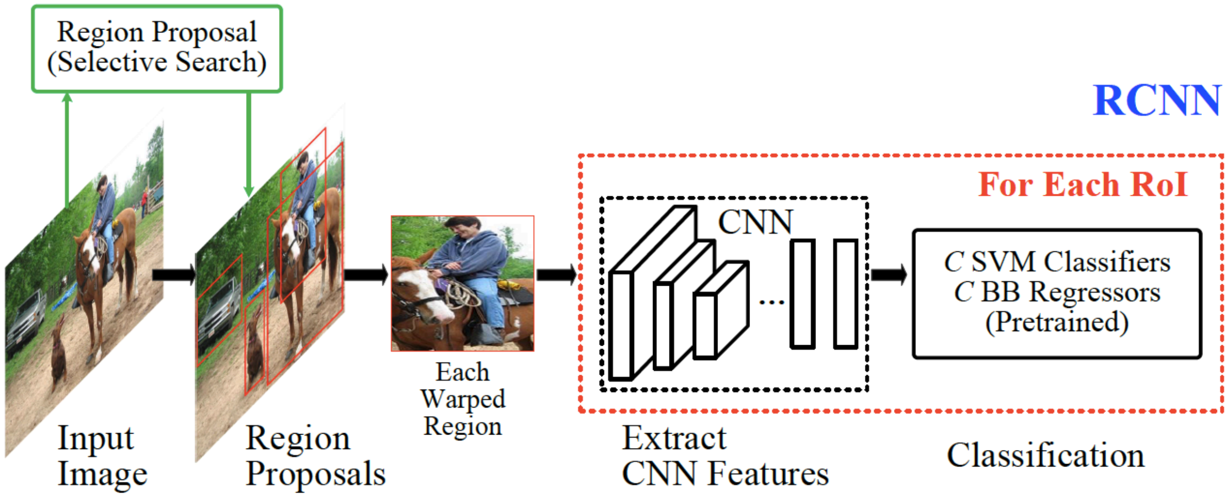
*source: https://arxiv.org/pdf/1809.02165.pdf

Table of Contents

Object Detection: R-CNN [Girshick et al., 2013]

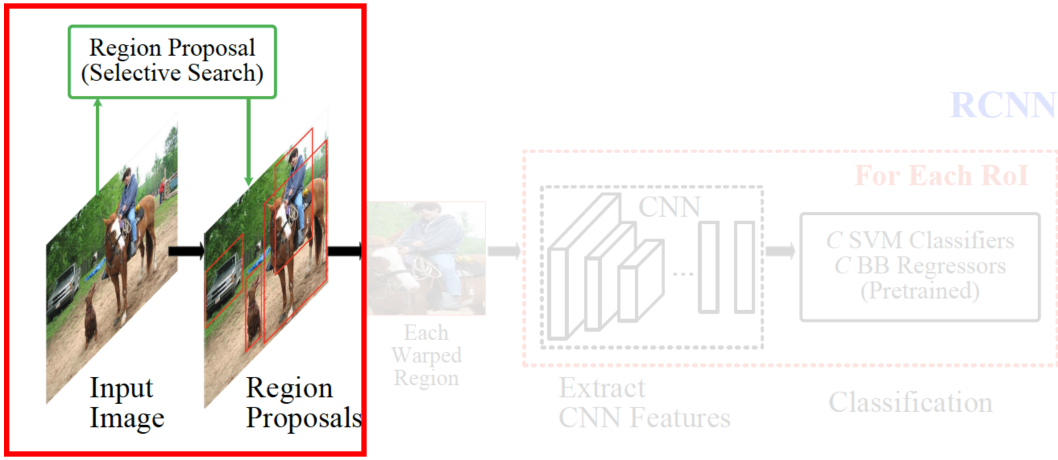
- Region-based Convolutional Network (**R-CNN**)
  - **First** to explore CNN in **object detection**
  - ILSVRC detection challenge winner in 2013
  - **Multi-stage** pipeline

- High-level diagrams

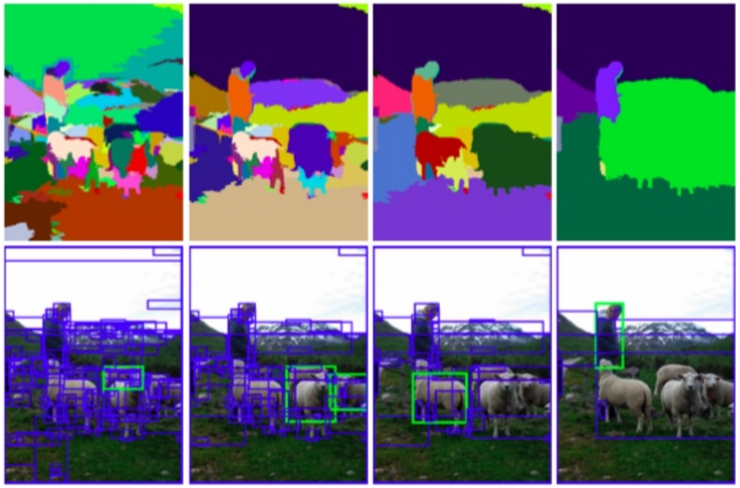


* source: https://arxiv.org/pdf/1809.02165.pdf
https://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html

Object Detection: R-CNN [Girshick et al., 2013]



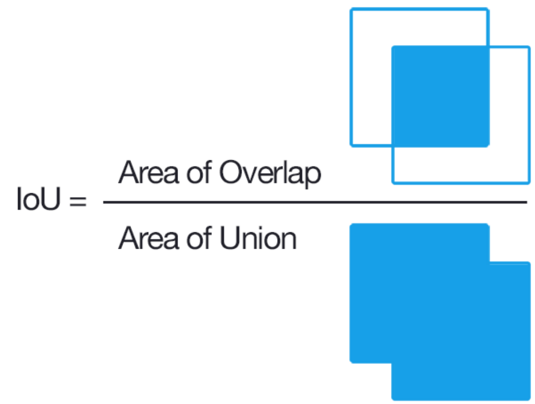
- Stage 1: **Region proposal**
    - Find candidate **regions** that might contain **objects**
    - Use selective search [Uijlings et al., 2013]



*source: https://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html

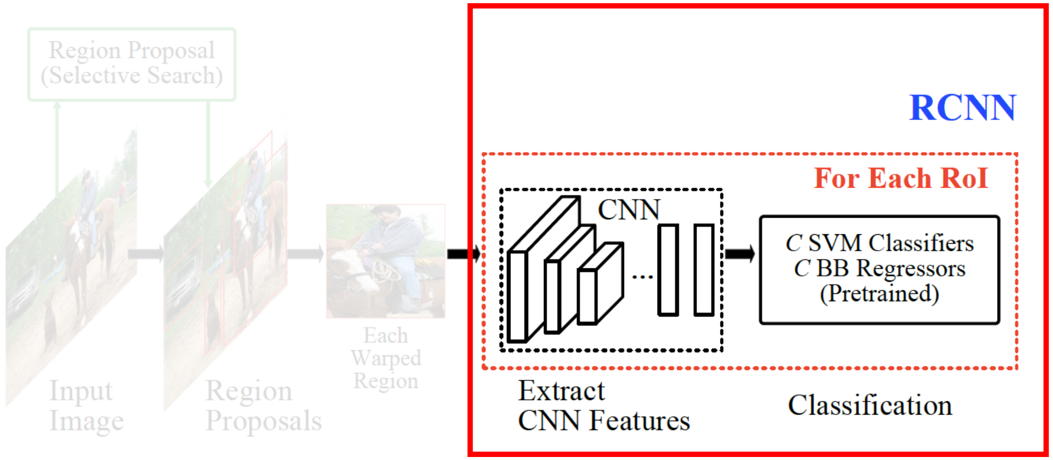
Object Detection: R-CNN [Girshick et al., 2013]

- Stage 2: **Fine-tine CNN**
  - Pre-train CNN (e.g., VGG-16) on ImageNet
  - **Fine-tune** CNN using **positive** samples (IoU > 0.5)
  - IoU: Intersection over union
  - Classify **N+1 classes** (N classes + background)

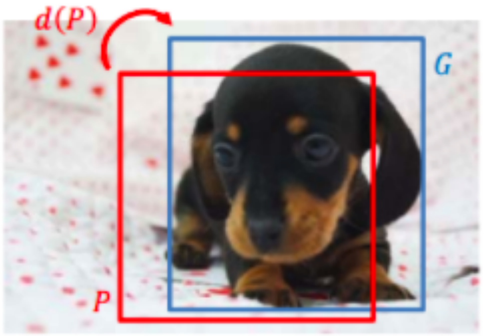


* source: https://arxiv.org/pdf/1809.02165.pdf
https://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html

Object Detection: R-CNN [Girshick et al., 2013]



- Stage 3: **Classification + bbox regression**
  - Using CNN features, train **N+1 SVMs** for each class for **binary classification**
  - Train bounding box regressors for refinement (mapping from region proposal P to ground truth G)

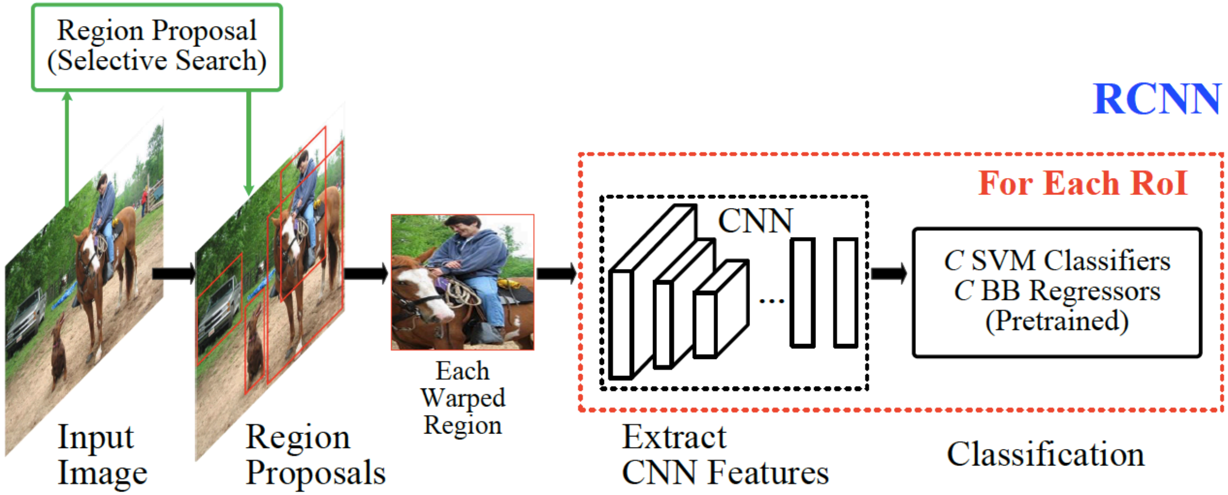


Bounding box regression example

* source: https://arxiv.org/pdf/1809.02165.pdf
https://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html

Object Detection: R-CNN [Girshick et al., 2013]

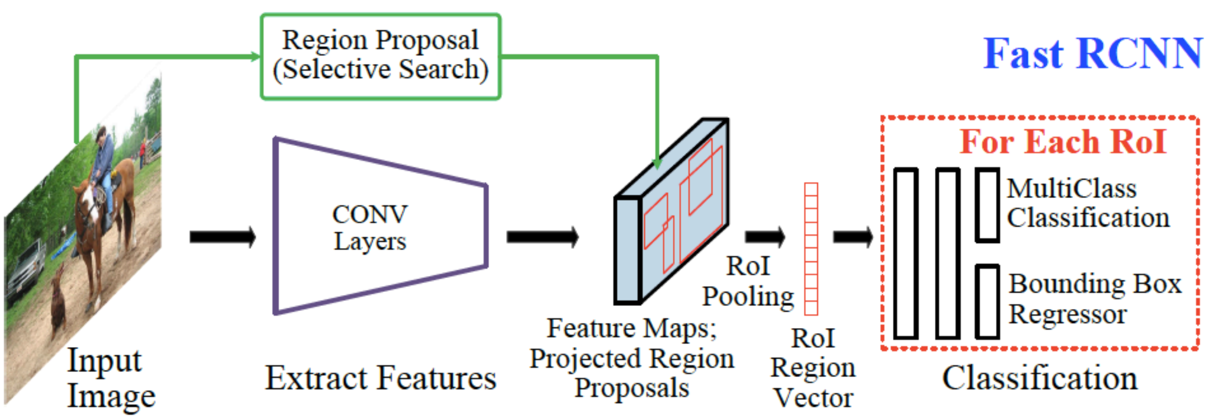
- Contributions
  - (+) **First** to explore CNN in **object detection**
  - (+) ILSVRC detection challenge winner in 2013
- Limitations
  - (-) **Slow** (need to compute output for every region proposal)
  - (-) **Complicated** multi-stage training scheme
  - (-) **CNN** features are **not updated** in response to SVMs and bbox regressors

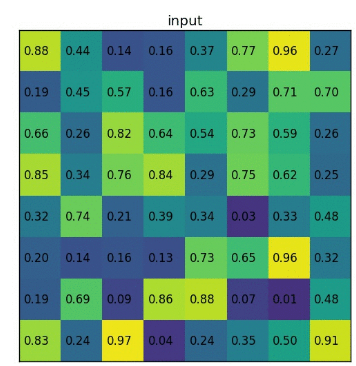


* source: https://arxiv.org/pdf/1809.02165.pdf
https://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html

Object Detection: Fast R-CNN [Girshick et al., 2015]

- Fast Region-based Convolutional Network (**Fast R-CNN**)
  - Better performance & Reduce computation time compared to R-CNN
  - ROI (Region of interest) pooling layer to output **fixed-size features** from each **region**
    - **Feature** map is **calculated only once** per each image
    - In previous R-CNN, need to calculate for all region proposals
  - Train softmax classifier + bounding box regressor on top
- Limitation
  - (-) Still uses **selective search** for region proposals to compute ROI features



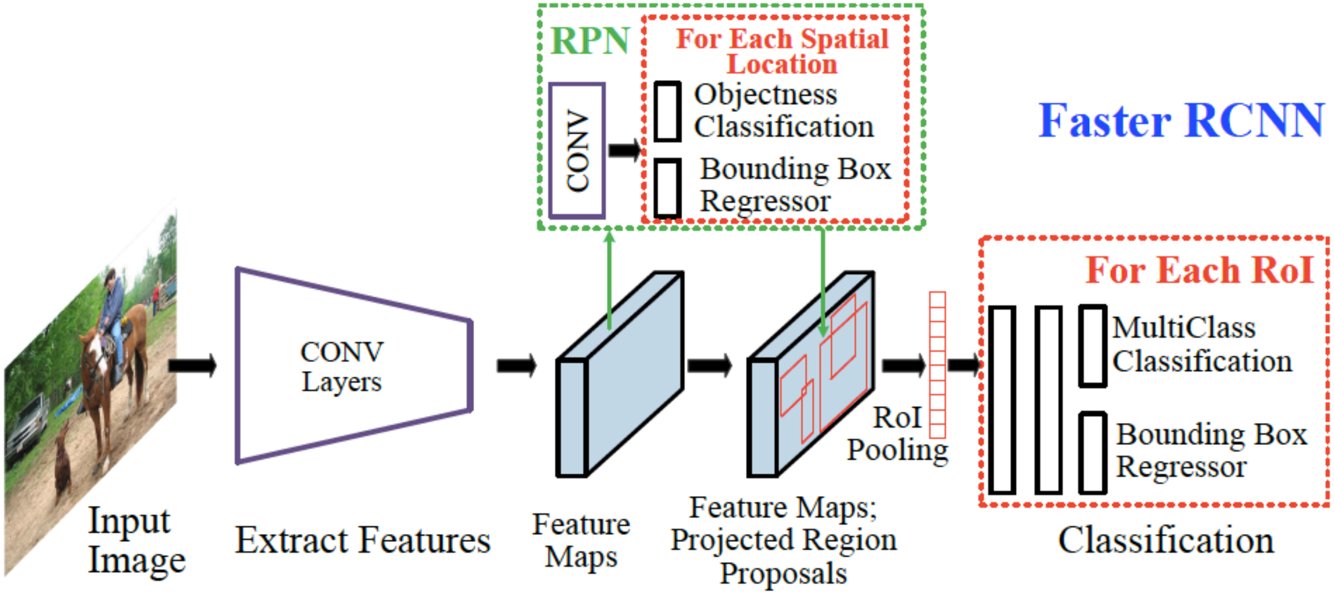


RoI Pooling layer

*source: https://arxiv.org/pdf/1504.08083.pdf
https://arxiv.org/pdf/1809.02165.pdf

Object Detection: Faster R-CNN [Shaoqing et al., 2016]

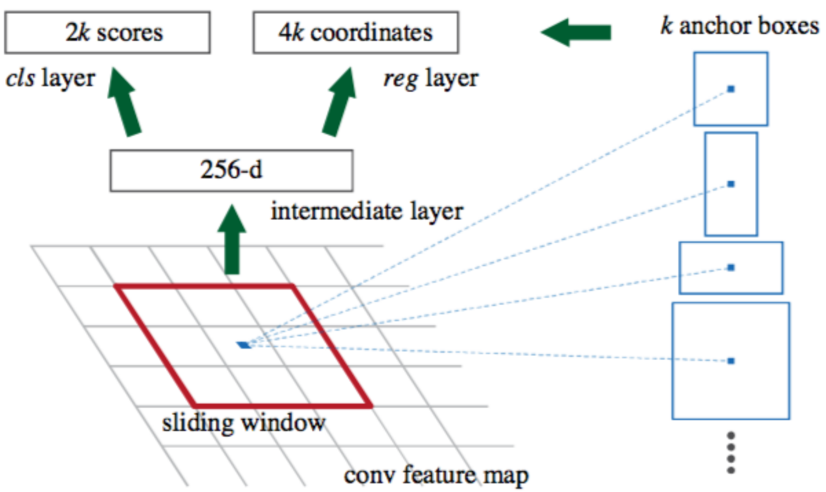
- Faster Region-based Convolutional Network (**Faster R-CNN**)
  - ILSVRC Detection challenge winner in 2015
  - Propose Region Proposal Network (**RPN**)
    - **Let CNN do region proposal** (no selective search)
  - Fast R-CNN + RPN = Faster R-CNN
  - **×34 faster** than Fast R-CNN (one of the trained models)



*source: https://arxiv.org/pdf/1504.08083.pdf
https://arxiv.org/pdf/1809.02165.pdf

Object Detection: Faster R-CNN [Shaoqing et al., 2016]

- More details of **RPN**
  - Resize every input image to have shorter side size of 600
  - Output k = 9 anchor boxes **per** each 3x3 **sliding window** in conv5 feature map
    - 3 different scales (128,256,512) x 3 different aspect ratios (1:1, 2:1, 1:2)
  - Use **NMS (Non Maximum Suppression)** to reduce overlapped boxes
  - Results in ~2000 bboxes per an image
  - Train classification + bbox regression on top using anchor boxes as reference

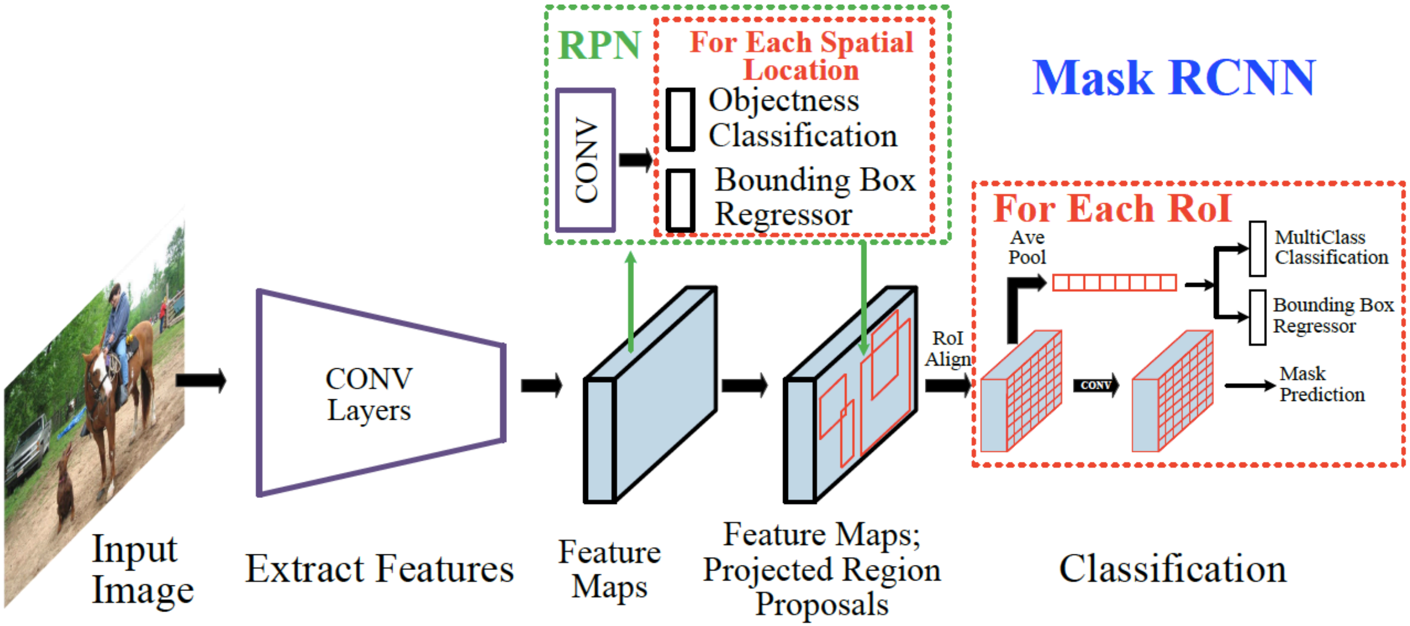


Region Proposal Network (RPN)

*source: https://arxiv.org/pdf/1504.08083.pdf
https://arxiv.org/pdf/1809.02165.pdf

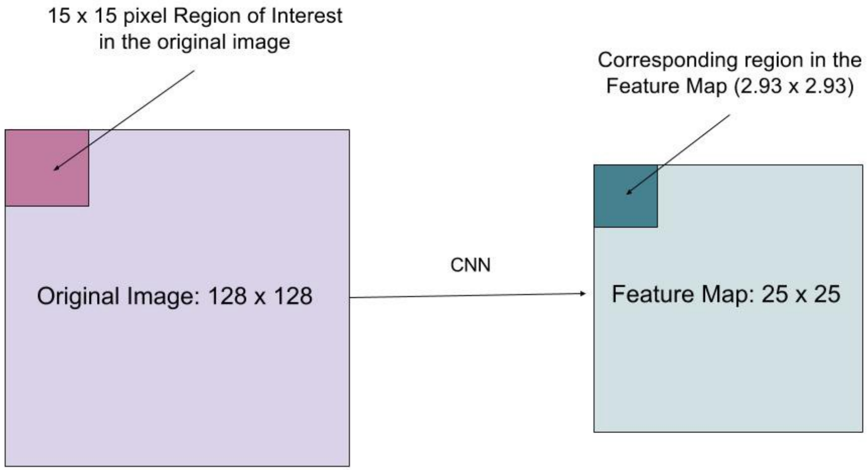
From Detection to Segmentation: Mask R-CNN [He et al., 2017]

- Can we extend the ideas used in detection model to pixel-level segmentation ?
  - **Mask R-CNN** extends Faster R-CNN to solve **both** detection+segmention
- Idea : Add **pathway** to predict **object mask** in parallel with box detection
  - **Input**: CNN feature maps
  - **Output**: A binary mask (a matrix with 1s on all locations where pixel belongs to the object and 0s elsewhere)



*source: https://arxiv.org/pdf/1703.06870.pdf
https://arxiv.org/pdf/1809.02165.pdf

- **ROI-Align**: Modification on ROI pooling layer for **better pixel-level alignment**
  - ex) original image size of 128×128, feature map size of 25×25
  - ROI of 15×15 $\quad \frac{15}{128}\sim\frac{2.93}{25}$
  - Corresponding region in feature map $\sim 2.93\times2.93$ pixels
  - Previous ROI-Pooling layer: round to 3×3 (0.5 pixel difference in the worst case)
  - **ROI-Align**
    - Bilinear interpolation to **precisely** estimate what would be in 2.93 pixels
    - Results in **better detection performance**



	AP	AP_{50}	AP_{75}	AP^{bb}
RoIPool	23.6	46.5	21.6	28.2
RoIAlign	**30.9**	**51.8**	**32.1**	**34.0**
	+7.3	*+ 5.3*	*+10.5*	*+5.8*

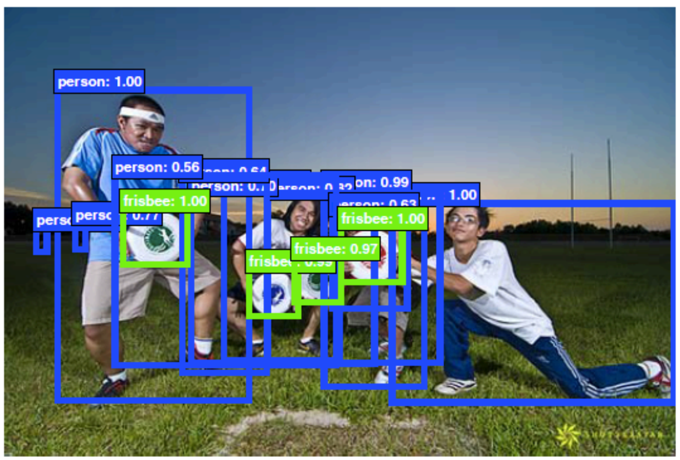
*source: https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4
https://arxiv.org/pdf/1703.06870.pdf

Mask R-CNN: Example results on MS COCO dataset

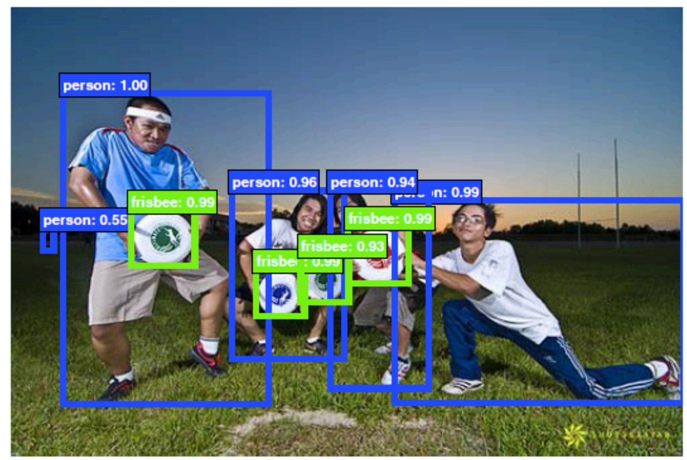
*source: https://arxiv.org/pdf/1703.06870.pdf

Object Detection: Cascade R-CNN [Cai et al., 2018]

- Detectors trained with **IoU threshold of 0.5** usually produces **noisy** results (a)
- How to train **high-quality** detectors?



(a) Detection of $u = 0.5$

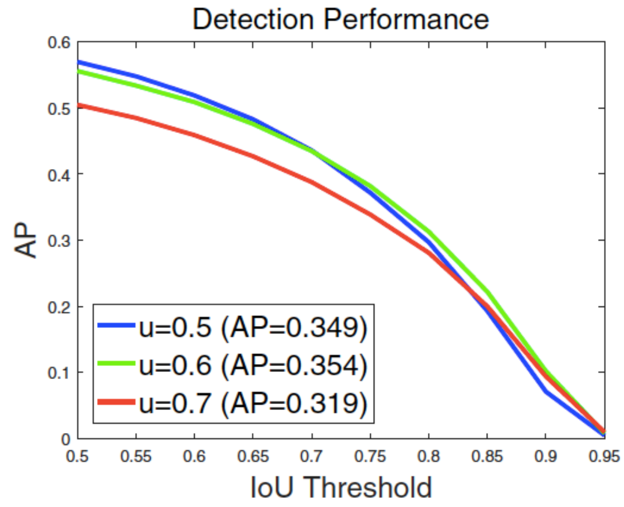


(b) Detection of $u = 0.7$

*source: https://arxiv.org/pdf/1712.00726.pdf

Object Detection: Cascade R-CNN [Cai et al., 2018]

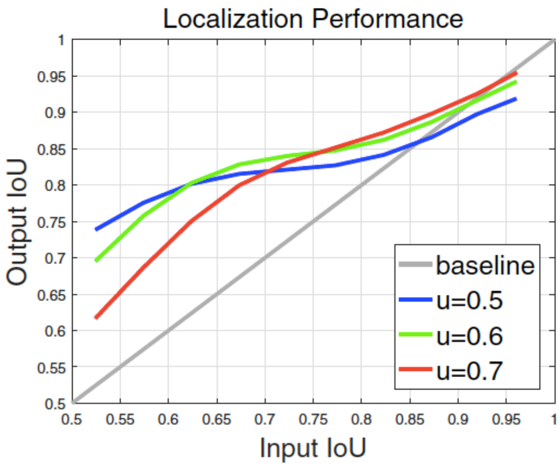
- Detectors trained with **IoU threshold of 0.5** usually produces **noisy** results
- How to train **high-quality** detectors?
  - Simply increasing threshold when training degrades performance:
  - Why?
    - Due to over-fitting
    - Number of **positive samples** largely **decrease** with large IoU threshold

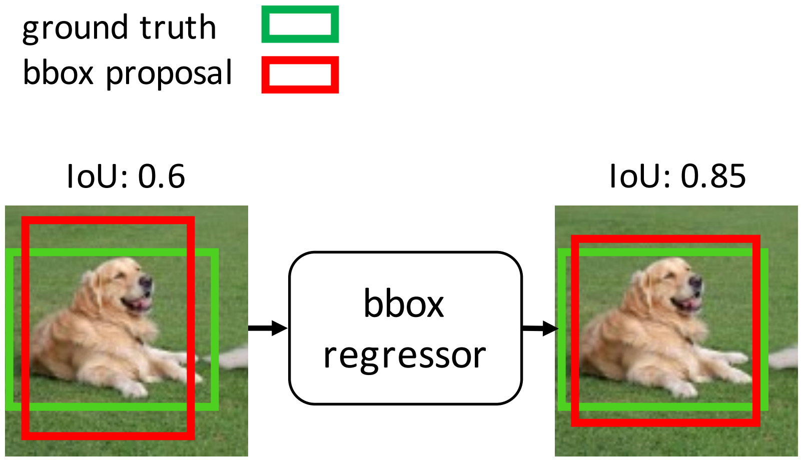


*source: https://arxiv.org/pdf/1712.00726.pdf

Object Detection: Cascade R-CNN [Cai et al., 2018]

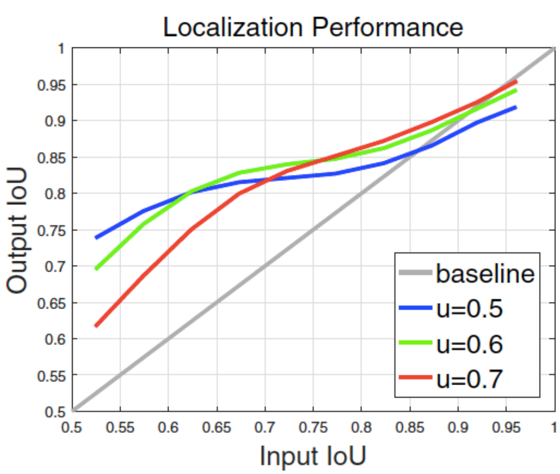
- Detectors trained with **IoU threshold of 0.5** usually produces **noisy** results
- How to train **high-quality** detectors?
  - Simply increasing threshold when training degrades performance:
  - Why?
    - Due to over-fitting
    - Number of **positive samples** largely **decrease** with large IoU threshold
  - Notice that the box regressor always produce better results than original input:

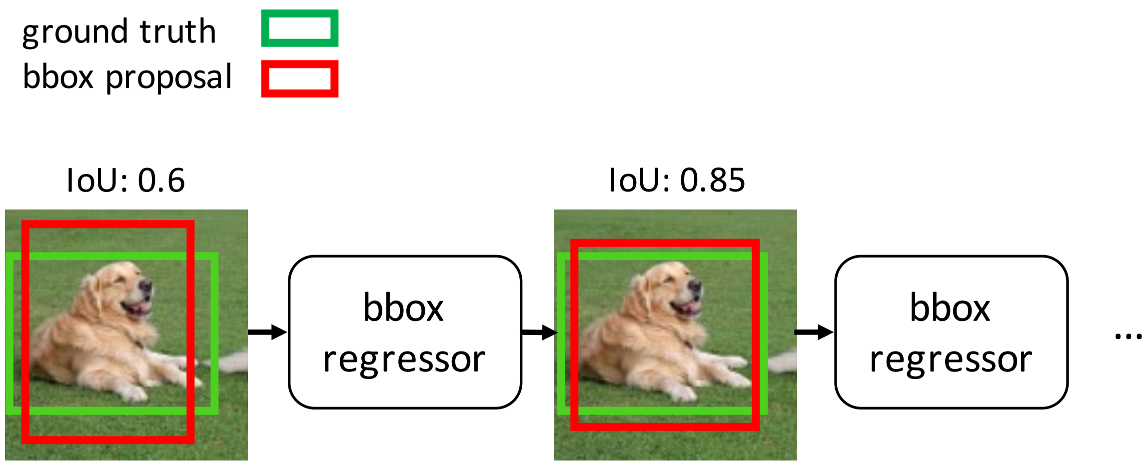




Object Detection: Cascade R-CNN [Cai et al., 2018]

- Detectors trained with **IoU threshold of 0.5** usually produces **noisy** results
- How to train **high-quality** detectors?
  - Simply increasing threshold when training degrades performance:
  - Why?
    - Due to over-fitting
    - Number of **positive samples** largely **decrease** with large IoU threshold
  - Notice that the box regressor always produce better results than original input:
- Idea: Using **cascade of detectors** with **increasing IoU threshold**

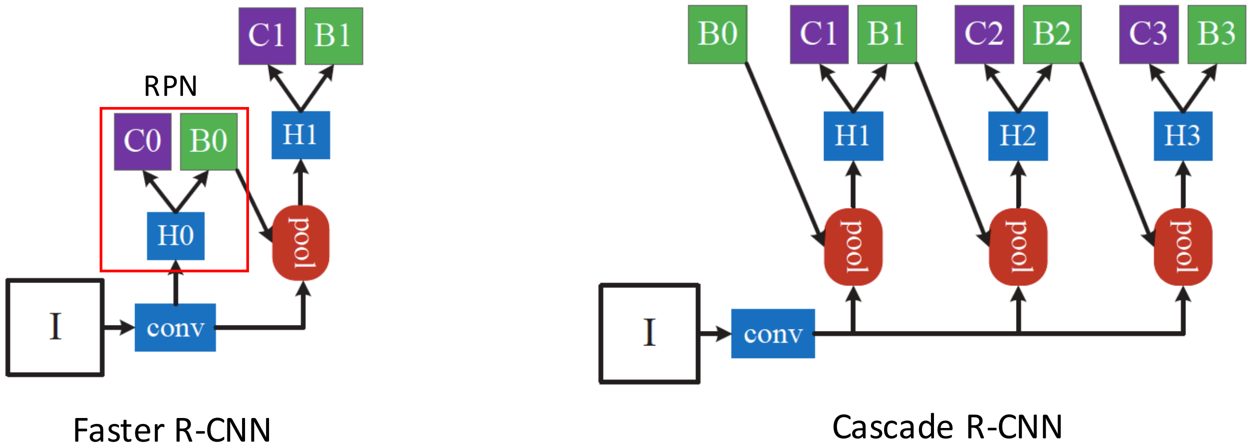




*source: https://arxiv.org/pdf/1712.00726.pdf

Object Detection: Cascade R-CNN [Cai et al., 2018]

- **Sequence** of **detectors** trained with **increasing IoU thresholds**
  - To be sequentially more selective against close false positives
  - **State-of-the-art** results compared to existing frameworks



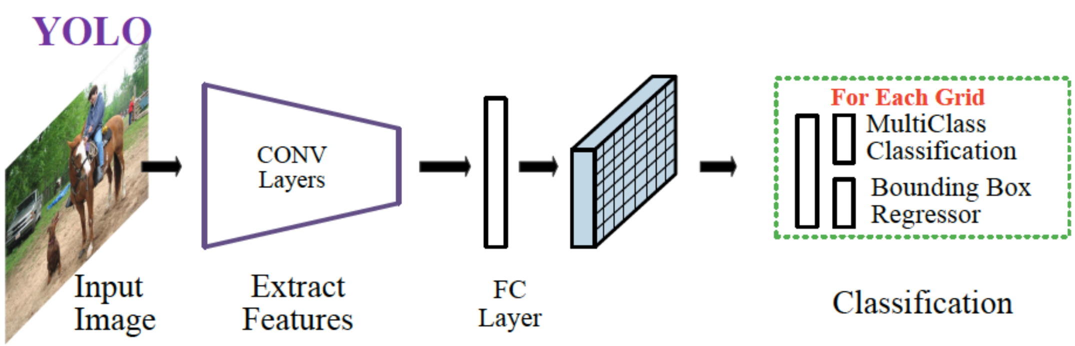
	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv2 [29]	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [25]	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
RetinaNet [24]	ResNet-101	39.1	59.1	42.3	21.8	42.7	50.2
Faster R-CNN+++ [18]*	ResNet-101	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [23]	ResNet-101	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN w FPN+ (ours)	ResNet-101	38.8	61.1	41.9	21.3	41.8	49.8
Faster R-CNN by G-RMI [19]	Inception-ResNet-v2	34.7	55.5	36.7	13.5	38.1	52.0
Deformable R-FCN [5]*	Aligned-Inception-ResNet	37.5	58.0	40.8	19.4	40.1	52.5
Mask R-CNN [16]	ResNet-101	38.2	60.3	41.7	20.1	41.1	50.2
AttractioNet [11]*	VGG16+Wide ResNet	35.7	53.4	39.3	15.6	38.0	52.7
Cascade R-CNN	ResNet-101	**42.8**	**62.1**	**46.3**	**23.7**	**45.5**	**55.2**

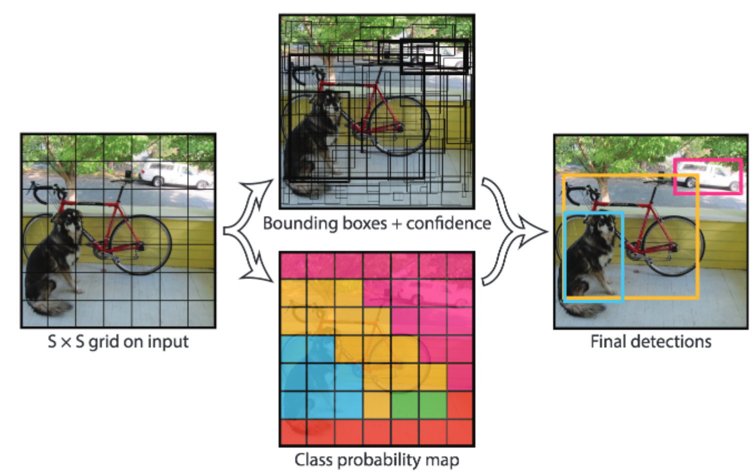
*source: https://arxiv.org/pdf/1712.00726.pdf

Table of Contents

Object Detection: You Only Look Once (YOLO) [Redmon et al., 2016]

- Predicts boxes & class probabilities with a **single network** in a **single evaluation**
- Object detection as **single regression problem**
  - Each image divided into $S \times S$ grid cell
  - B bounding boxes are predicted (regression) with a **confidence score**
  - A most likely class is predicted among C classes (per each grid cell)
  - Final output size: $S \times S \times (B \times 5 + C)$
  - **NMS (Non Maximum Supression)**: Merge highly overlapped boxes





*source: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf
https://arxiv.org/pdf/1809.02165.pdf

Object Detection: YOLO [Redmon et al., 2016]

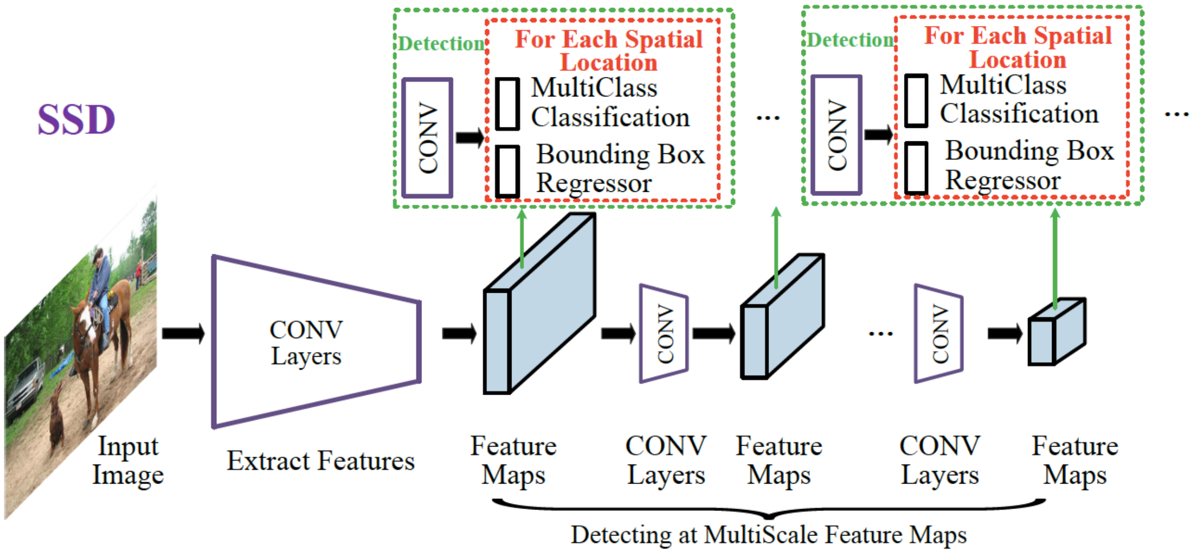
- You Only Look Once (**YOLO**)
  - (+) See entire image as input (better catch global context)
  - (+) **Very Fast**
  - (-) Difficulty in predicting small objects in groups
  - (-) **Accuracy trade-off** with **speed**

Model	mAP	FPS	Real Time speed
Fast YOLO	52.7%	**155**	Yes
YOLO	**63.4%**	45	Yes
YOLO VGG-16	66.4%	21	No
Fast R-CNN	70.0%	0.5	No
Faster R-CNN VGG-16	73.2%	7	No
Faster R-CNN ZF	62.1%	18	No

*source: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf

Object Detection: Single-shot Multibox Detector (SSD) [Wei et al., 2017]

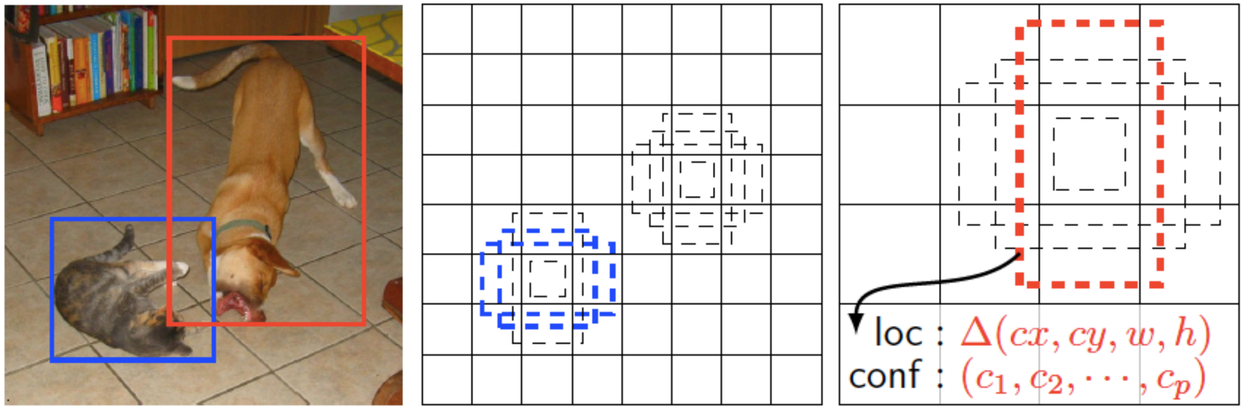
- Goal: As **fast** as YOLO while being **accurate** as Faster-RCNN
- Key ideas
  - Use **multi-scale features** instead of using single layer
  - Use **default anchor box** per each multi-scale feature grid (similar to RPN)
  - **Hard negative mining**: reduce imbalance between negative and positive samples



*source: https://arxiv.org/pdf/1512.02325.pdf
https://arxiv.org/pdf/1809.02165.pdf

Object Detection: Single-shot Multibox Detector (SSD) [Wei et al., 2017]

- Goal: As **fast** as YOLO while being **accurate** as Faster-RCNN
- Key ideas
  - Use **multi-scale features** instead of using single layer
  - Use **default anchor box** per each multi-scale feature grid (similar to RPN)
  - **Hard negative mining**: reduce imbalance between negative and positive samples



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

*source: https://arxiv.org/pdf/1512.02325.pdf

Object Detection: Single-shot Multibox Detector (SSD) [Wei et al., 2017]

- Effect of **multi-scale** features

Prediction source layers from:						mAP use boundary boxes?		# Boxes
conv4_3	conv7	conv8_2	conv9_2	conv10_2	conv11_2	Yes	No	
✔	✔	✔	✔	✔	✔	74.3	63.4	8732
✔	✔	✔	✔	✔		**74.6**	63.1	8764
✔	✔	✔	✔			73.8	68.4	8942
✔	✔	✔				70.7	69.2	9864
✔	✔					64.2	64.4	9025
	✔					62.4	64.0	8664

- As **fast** as YOLO, while being more **accurate** than Faster-RCNN

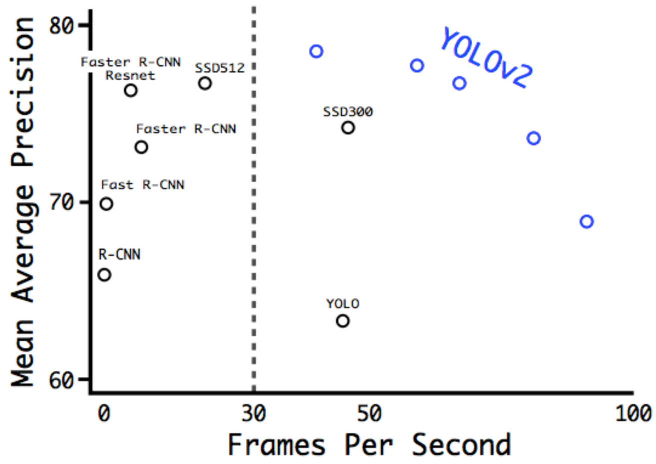
Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	$\sim 1000 \times 600$
Fast YOLO	52.7	155	1	98	448×448
YOLO (VGG16)	66.4	21	1	98	448×448
SSD300	74.3	46	1	8732	300×300
SSD512	76.8	19	1	24564	512×512
SSD300	74.3	59	8	8732	300×300
SSD512	76.8	22	8	24564	512×512

Performance compared to other detectors

*source: https://arxiv.org/pdf/1512.02325.pdf

Object Detection: YOLOv2 [Redmon et al., 2017]

- Focus on **improving accuracy** while still being **fast**
- Modifications on YOLO
  - Higher resolution images
  - Batch Normalization
  - Final FC layer is removed
  - New network , multi-scale
  - …



	YOLO								YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓	✓
anchor boxes?				✓	✓				
new network?					✓	✓	✓	✓	✓
dimension priors?						✓	✓	✓	✓
location prediction?						✓	✓	✓	✓
passthrough?							✓	✓	✓
multi-scale?								✓	✓
hi-res detector?									✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

Path from YOLO to YOLOv2 on VOC2007 dataset.

*source: http://openaccess.thecvf.com/content_cvpr_2017/papers/Redmon_YOLO9000_Better_Faster_CVPR_2017_paper.pdf

Object Detection: YOLOv3 [Redmon et al., 2018]

- **YOLOv3**: An **incremental** improvement
  - Improve **accuracy** of YOLOv2 while still being **fast**
  - Better backbone architecture
  - K-means clustering to determine bounding box priors (3 different scales)

- Demo
  - https://www.youtube.com/watch?v=MPU2HistivI

Table of Contents

What is Visual Question Answering?

- **Visual Question Answering (VQA)**
  - Given **an image** and **a question related to the image**,
  - **Answer the question**

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

What is Visual Question Answering?

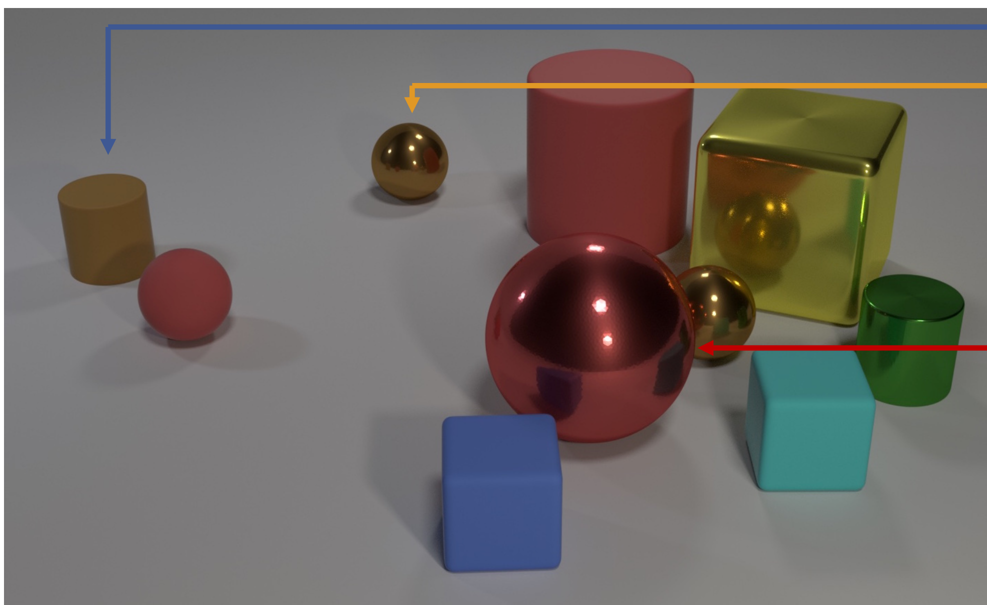
- **Visual Question Answering (VQA)**
  - Given **an image** and **a question related to the image**,
  - **Answer the question**

- Challenges
  - Need to understand the question
    - What kind of question? (e.g., yes/no, counting, comparison, …)
  - Need to understand objects in the given image
    - Object's attributes (e.g., color, shape, …)
    - Relation between objects (e.g., larger/smaller, left/right, …)
  - **Need to connect the question and image**
    - **Relation between words in question and objects in image**

CLEVR: A Benchmark Dataset for VQA

- CLEVR [Johnson et al., 2017] is a synthetic diagnostic dataset for language and visual reasoning
  - Attributes (color, shape, size) and positions of objects are randomly generated
  - Types of questions are counting, comparison, attribute identification, and so on.
  - To answer, **understanding natural languages and visual reasoning are required**

- Example:



Q) What size is the cylinder that is left of brown metal thing that is left of the big sphere?

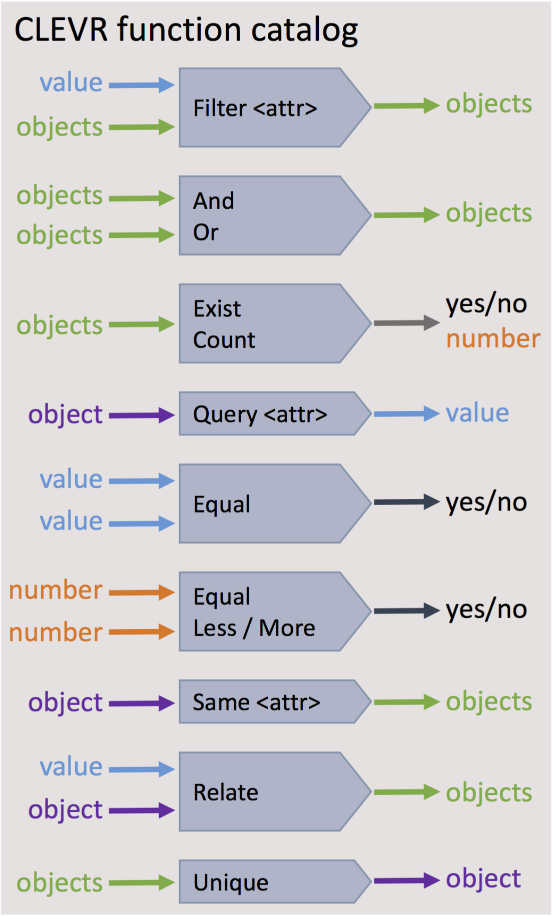
require reasoning to answer:

A) Small

*source: https://cs.stanford.edu/people/jcjohns/clevr/

How to generate questions?

- Questions in CLEVR are generated as functional programs
  1. Build a structure using pre-defined functions
  2. Generate the corresponding natural language question

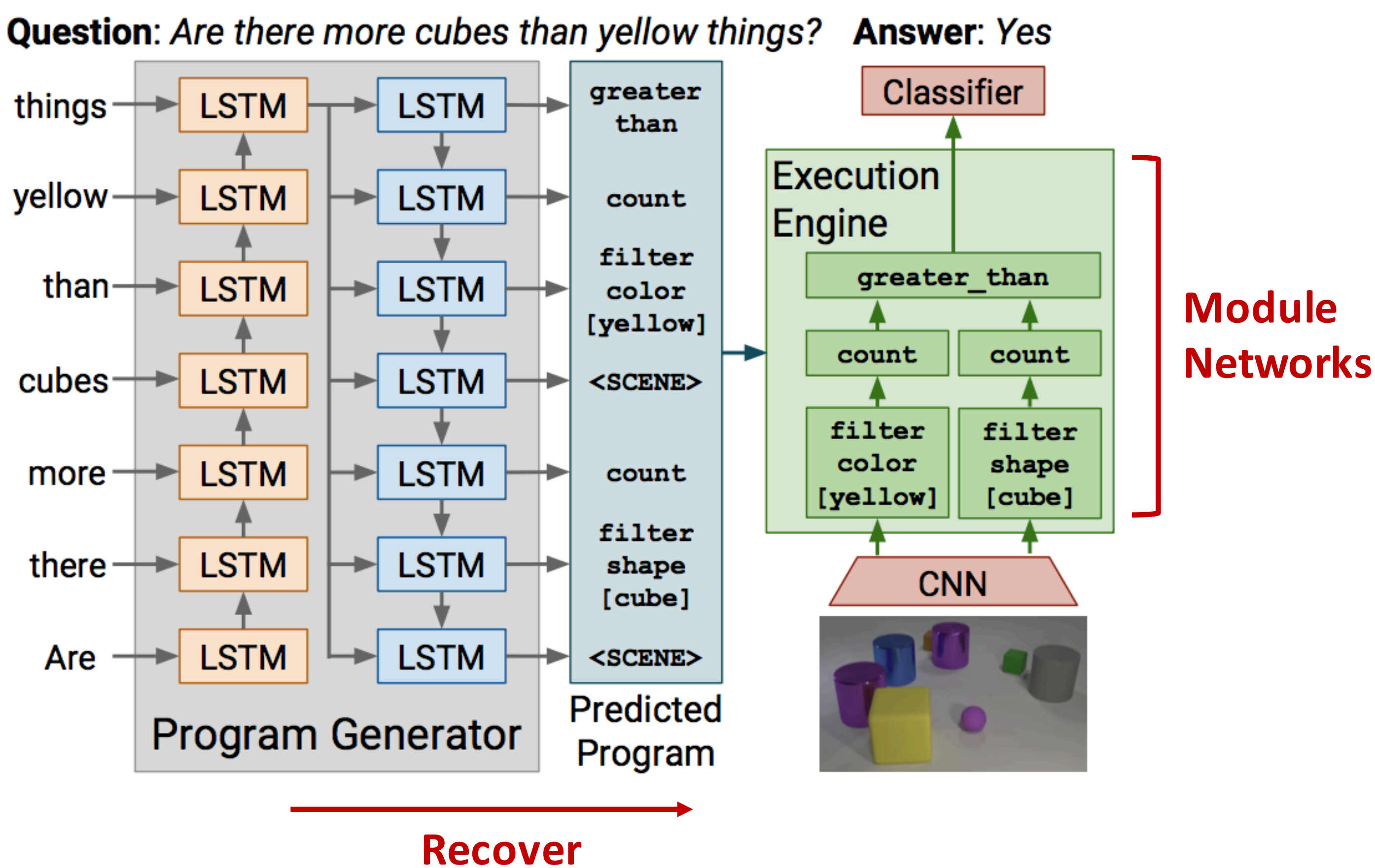


*source: https://cs.stanford.edu/people/jcjohns/clevr/

Table of Contents

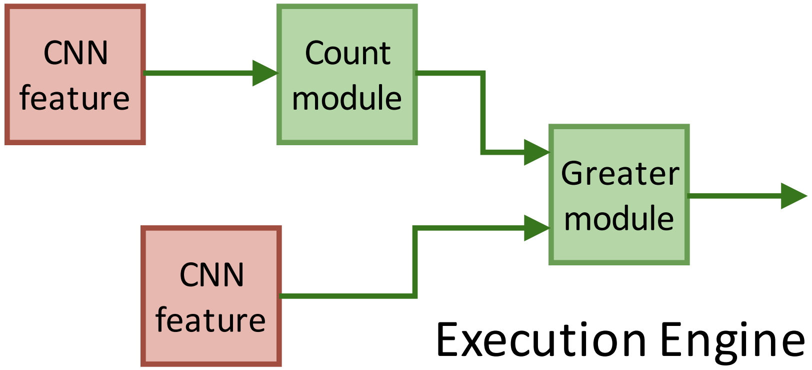
Inferring and Executing Programs for Visual Reasoning [Johnson et al., 2017]

- Idea
  - **Functional programs say how to understand/answer the question**
  - Recover the functional program from the question using RNN
  - Build a neural network based on the structure of the program



Inferring and Executing Programs for Visual Reasoning [Johnson et al., 2017]

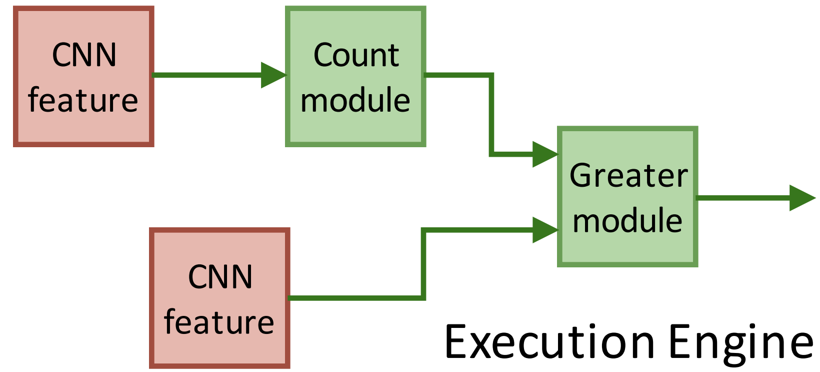
- **Neural Network Module**
  - **Each NN module** corresponds to one function in CLEVR catalog
  - Each module receives inputs from **CNN's features** or **outputs of other modules**



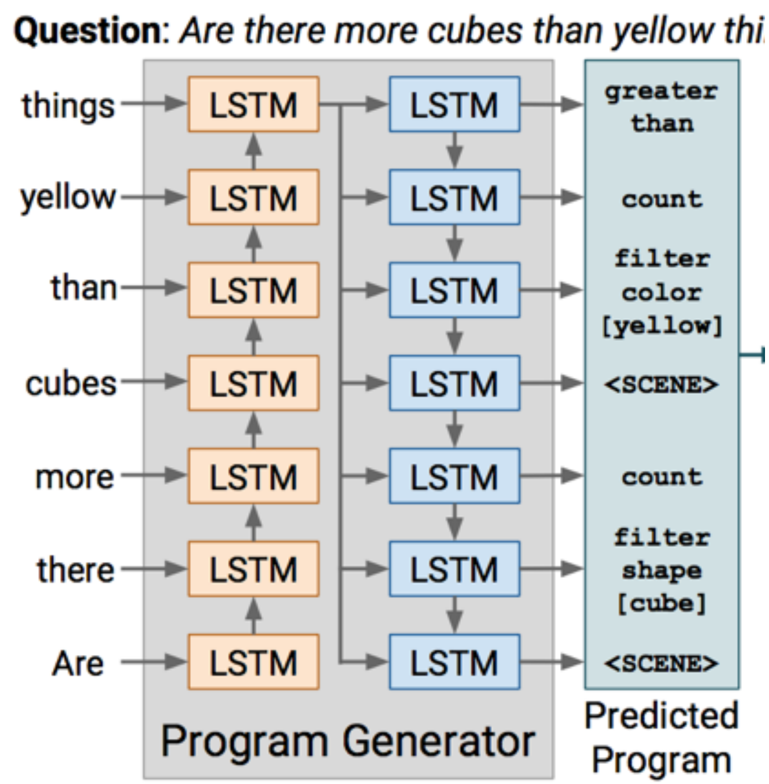
Execution Engine

Inferring and Executing Programs for Visual Reasoning [Johnson et al., 2017]

- **Neural Network Module**
  - **Each NN module** corresponds to one function in CLEVR catalog
  - Each module receives inputs from **CNN's features** or **outputs of other modules**



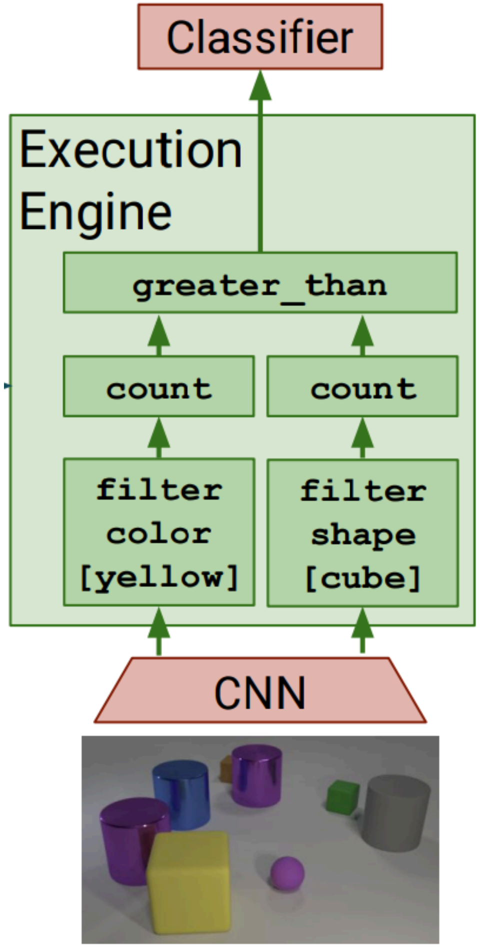
- **Program Generator (PG)**
  - PG outputs **a sequence of modules** given a question
  - The sequence is a prefix representation
  - PG can be **trained using ground-truth programs** for questions
    - CLEVR has 700k question-program pairs



*source: https://arxiv.org/pdf/1705.03633.pdf

Inferring and Executing Programs for Visual Reasoning [Johnson et al., 2017]

- **Execution Engine (EE)**
  - A combination of modules represents a neural network
  - CNN is a pre-trained network
  - **EE can be trained when the structure is fixed**

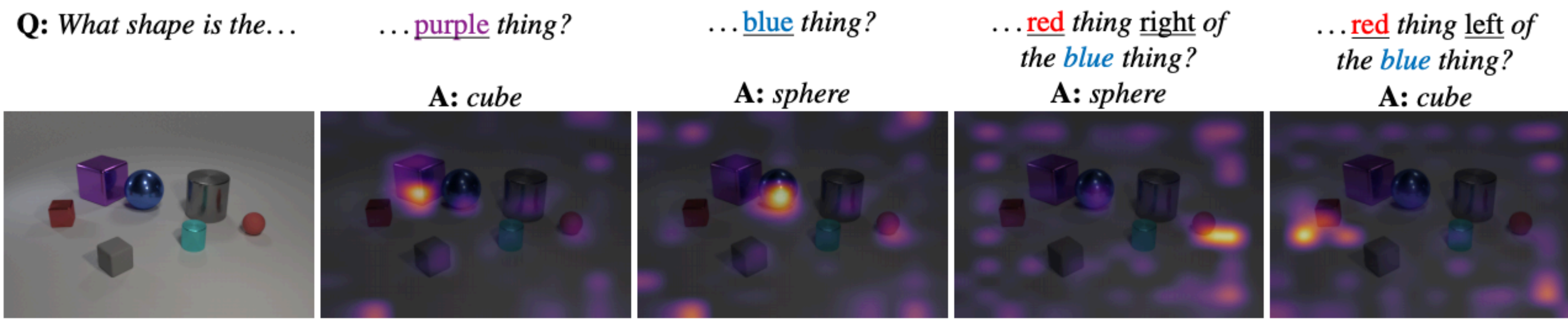


- Training Phases
  1. Train Program Generator
  2. Train Execution Engine with fixed PG
  3. Jointly fine-tune PG and EE via REINFORCE algorithm
     - PG is a **policy network**
     - The accuracy of EE is a **reward**

*source: https://arxiv.org/pdf/1705.03633.pdf

Inferring and Executing Programs for Visual Reasoning [Johnson et al., 2017]

- Changing modules affect visual attention and prediction of Execution Engine



- PG + EE outperforms existing baselines

			Compare Integer			Query				Compare				
Method	Exist	Count	Equal	Less	More	Size	Color	Mat.	Shape	Size	Color	Mat.	Shape	Overall
Q-type mode	50.2	34.6	51.4	51.6	50.5	50.1	13.4	50.8	33.5	50.3	52.5	50.2	51.8	42.1
LSTM	61.8	42.5	63.0	73.2	71.7	49.9	12.2	50.8	33.2	50.5	52.5	49.7	51.8	47.0
CNN+LSTM	68.2	47.8	60.8	74.3	72.5	62.5	22.4	59.9	50.9	56.5	53.0	53.8	55.5	54.3
CNN+LSTM+SA [46]	68.4	57.5	56.8	74.9	68.2	90.1	83.3	89.8	87.6	52.1	55.5	49.7	50.9	69.8
CNN+LSTM+SA+MLP	77.9	59.7	60.3	83.7	76.7	85.4	73.1	84.5	80.7	72.3	71.2	70.1	69.7	73.2
Human† [19]	96.6	86.7	79.0	87.0	91.0	97.0	95.0	94.0	94.0	94.0	98.0	96.0	96.0	92.6
Ours-strong (700K prog.)	**97.1**	**92.7**	**98.0**	**99.0**	**98.9**	**98.8**	**98.4**	**98.1**	**97.3**	**99.8**	**98.5**	**98.9**	**98.4**	**96.9**
Ours-semi (18K prog.)	95.3	90.1	93.9	97.1	97.6	98.1	97.1	97.7	96.6	99.0	97.6	98.0	97.3	95.4
Ours-semi (9K prog.)	89.7	79.7	85.2	76.1	77.9	94.8	93.3	93.1	89.2	97.8	94.5	96.6	95.1	88.6

*source: https://arxiv.org/pdf/1705.03633.pdf

Inferring and Executing Programs for Visual Reasoning [Johnson et al., 2017]

- **Limitations**
  - Require an assumption about questions, functional programs
    - i.e., require strong prior knowledge
  - Require additional supervision (program-question pairs) for training PG

- **Simple, but effective approaches for existing architectures without strong priors**
  - Modulating visual processing by language: MODERN, FiLM
  - Relational reasoning: Relation Network

Table of Contents

A Simple Neural Network Module for Relational Reasoning [Santoro et al., 2017]

- Motivation
  - The ability to **reason about relations between objects** is crucial
  - Many architectures do not focus explicitly on relational reasoning

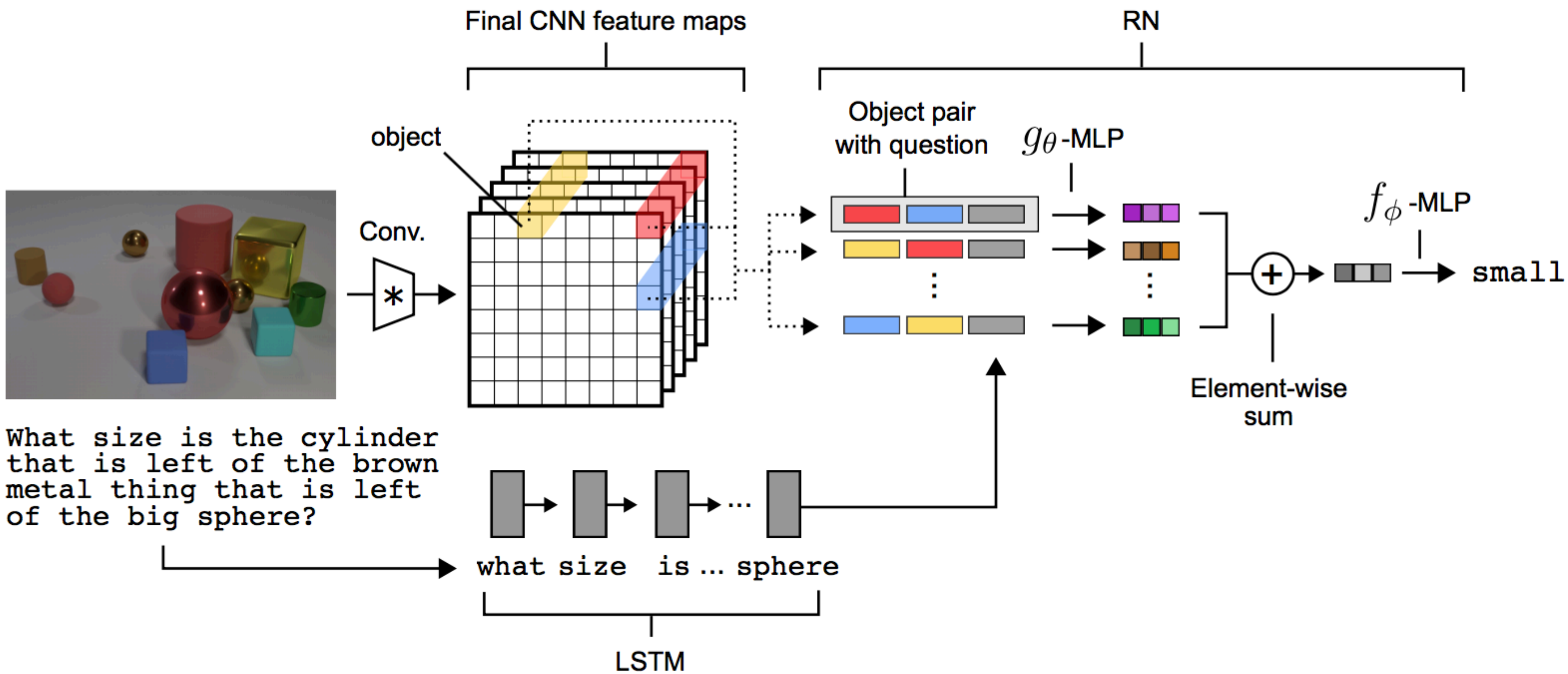
- **Relation Networks (RN)**

$$\mathrm{RN}(O) = f_\phi\left(\sum_{i,j} g_\theta(o_i, o_j)\right)$$

  - $O = \{o_1, \ldots, o_n\}$ and o_i is i-th object's representation (e.g., CNN features)
  - f_ϕ and g_θ are arbitrary functions (e.g., MLP)
  - **Strength:** (1) RNs learn to infer relations (2) RNs are data efficient (3) RNs operate on a set of objects

A Simple Neural Network Module for Relational Reasoning [Santoro et al., 2017]

- **RN-augmented CNN**

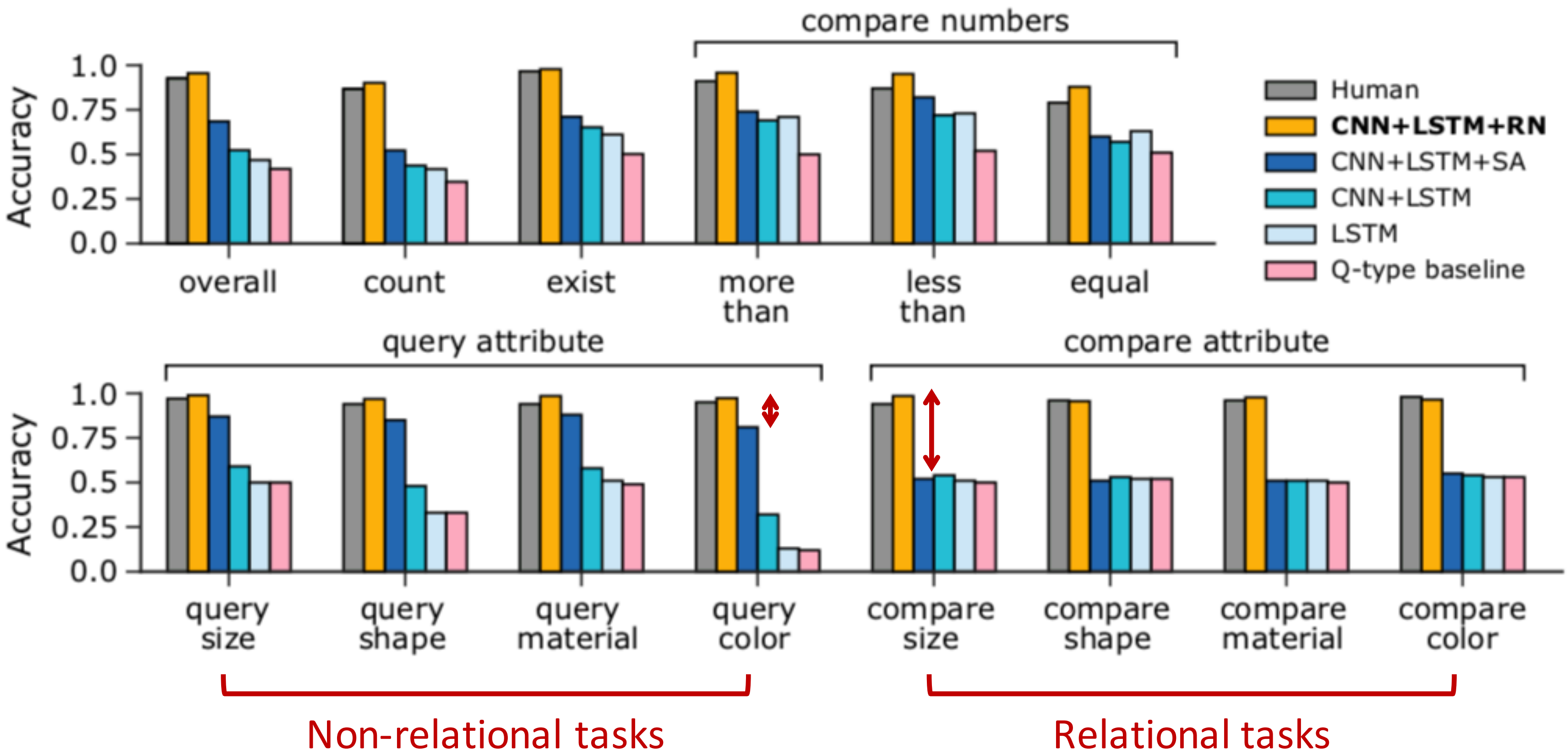


- Each pixel in final CNN features represents an object
- RN is conditioned on question embeddings (e.g., RNN hidden vectors)

*source: https://arxiv.org/pdf/1706.01427.pdf

A Simple Neural Network Module for Relational Reasoning [Santoro et al., 2017]

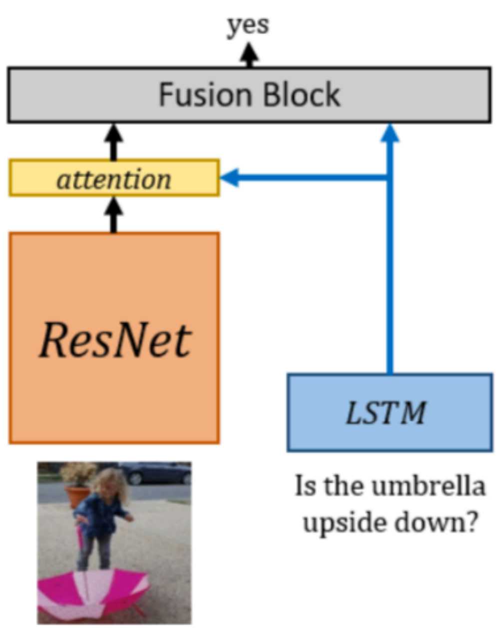
- RNs significantly outperforms other baselines **on relational tasks**



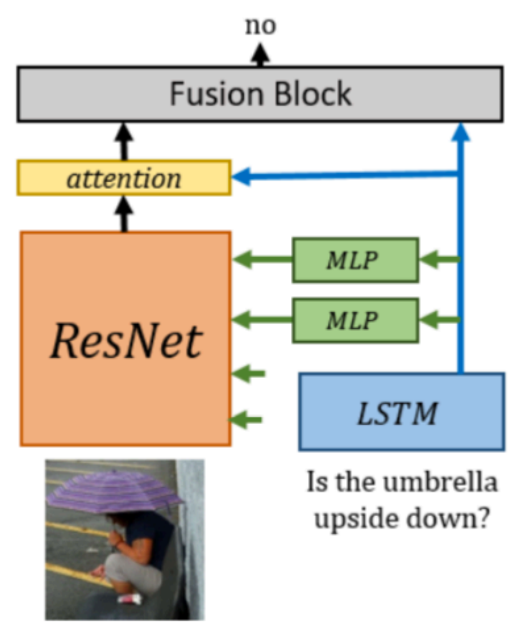
*source: https://arxiv.org/pdf/1706.01427.pdf

Modulating Early Visual Processing by Language [de Vries et al., 2017]

- Motivation
  - Prior works use features obtained from pre-trained CNNs
  - However, **depending on linguistic inputs** (questions), **the visual processing** (CNN layers) **should be changed**
  - How to modulate visual processing based on linguistic inputs?



Previous approach



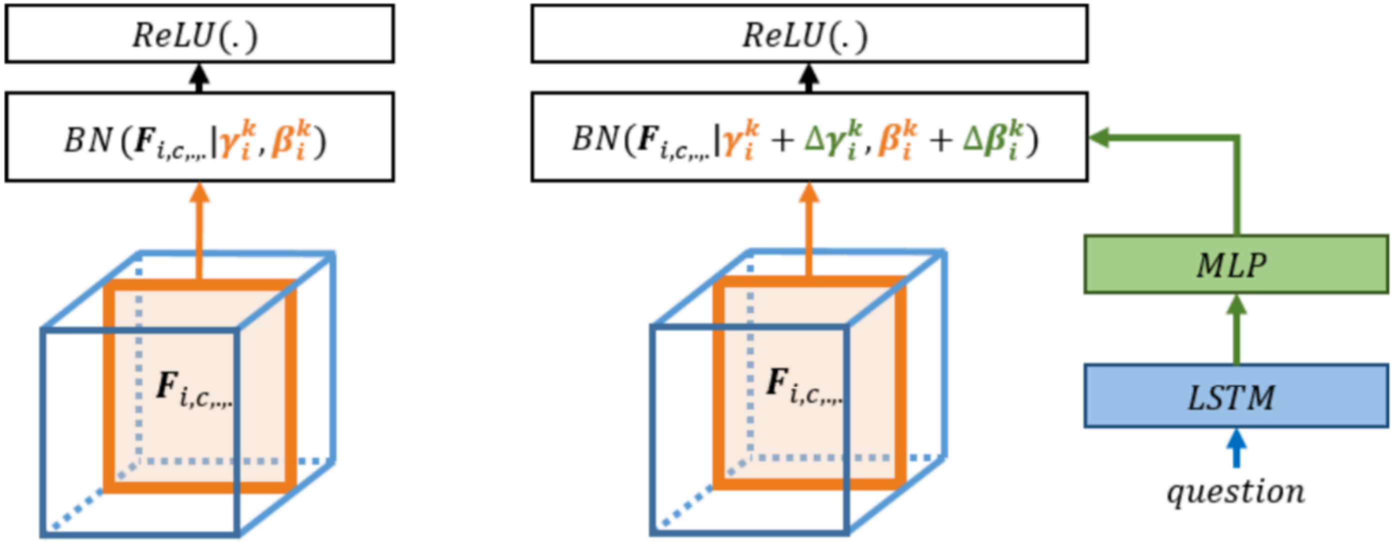
MODERN (this paper)

*source: https://arxiv.org/pdf/1707.00683.pdf

- **Conditional Batch Normalization (CBN)**
  - CBN modulates **affine parameters** in BN using embedding vectors from LSTM

$$\mathrm{BN}(x|\gamma,\beta) \rightarrow \mathrm{BN}(x|\gamma+\Delta\gamma,\beta+\Delta\beta)$$

where $(\Delta\gamma,\Delta\beta) = \mathrm{MLP}(\mathrm{LSTM}(\mathrm{question}))$

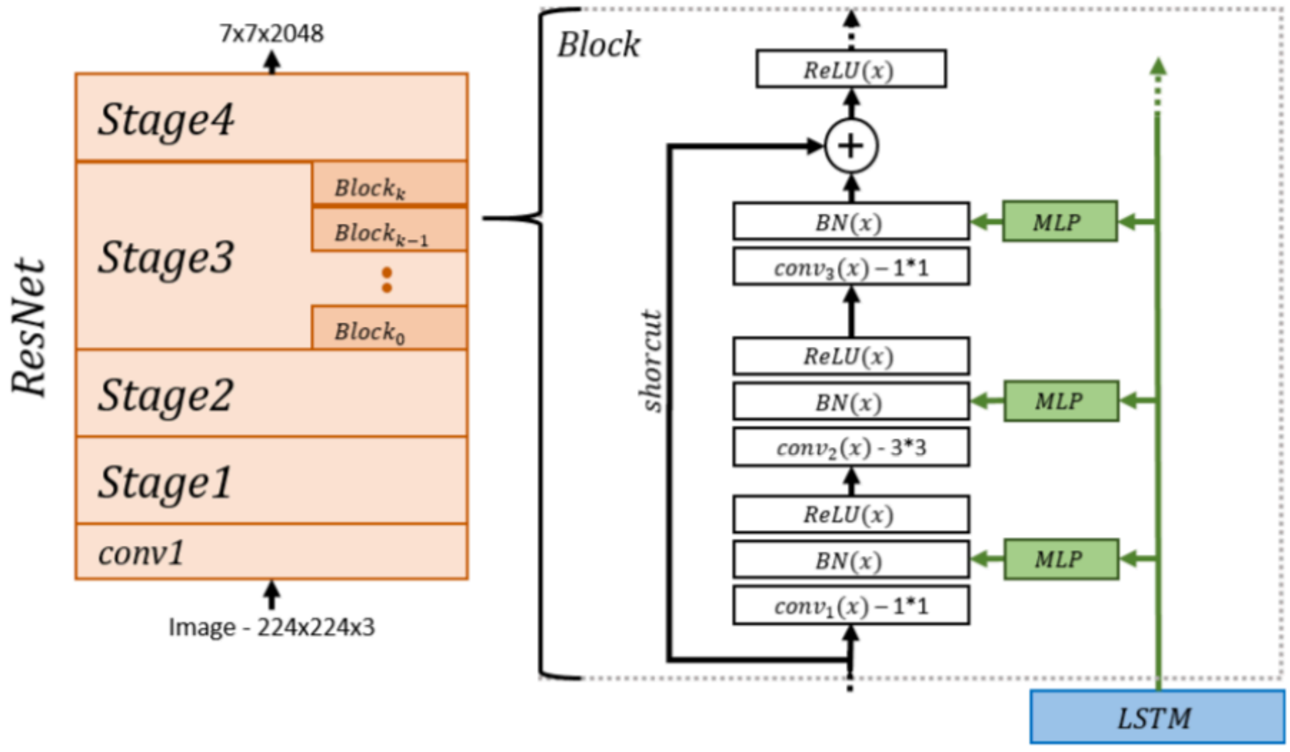


*source: https://arxiv.org/pdf/1707.00683.pdf

- **Conditional Batch Normalization (CBN)**
  - CBN modulates **affine parameters** in BN using embedding vectors from LSTM

$$\mathrm{BN}(x|\gamma,\beta) \rightarrow \mathrm{BN}(x|\gamma+\Delta\gamma,\beta+\Delta\beta)$$

where $(\Delta\gamma, \Delta\beta) = \mathrm{MLP}(\mathrm{LSTM}(\text{question}))$

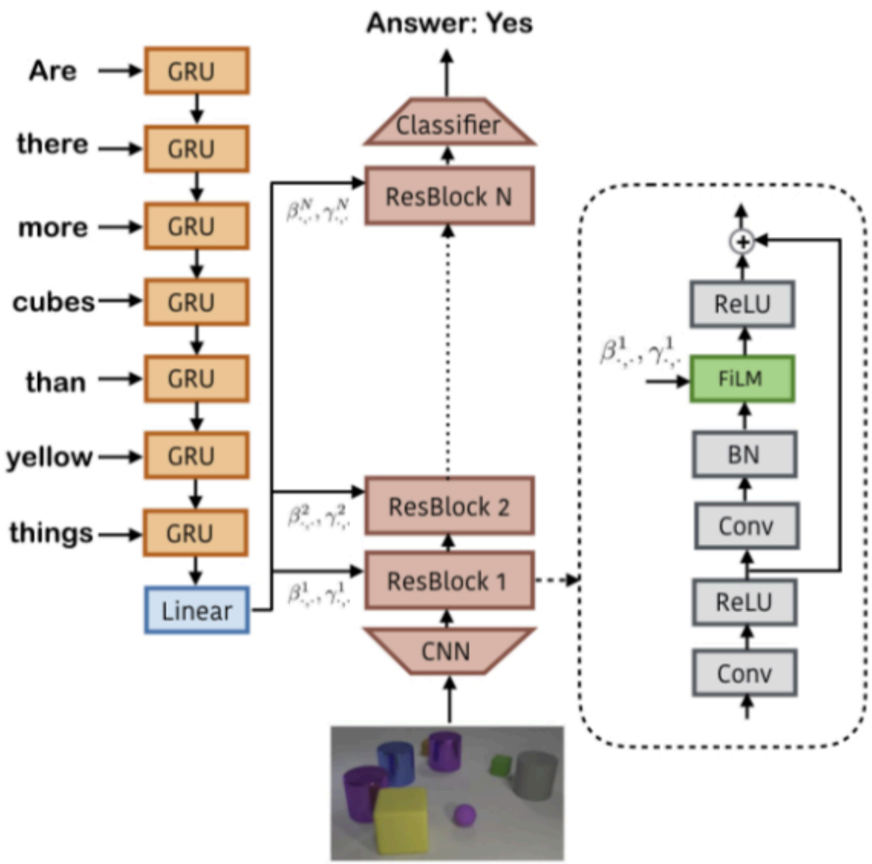


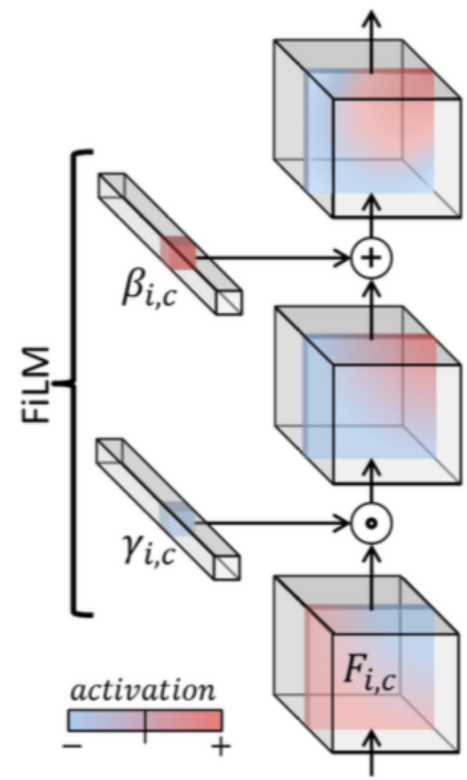
*source: https://arxiv.org/pdf/1707.00683.pdf

Modulating Early Visual Processing by Language [de Vries et al., 2017]

- **Feature-wise Linear Modulation (FiLM)** [Perez et al., 2018]
  - **Affine transformation of features** instead of modulate affine parameters in BN

$$\mathrm{FiLM}(x|\gamma,\beta) = \gamma x + \beta \quad \text{where} \quad (\gamma,\beta) = f(\text{question})$$





*source: https://arxiv.org/pdf/1709.07871.pdf

Modulating Early Visual Processing by Language [de Vries et al., 2017]

- FiLM also changes **visual attention** depending on questions

Q: *What shape is the...* | *...purple thing?* **A:** *cube* | *...blue thing?* **A:** *sphere* | *...red thing right of the blue thing?* **A:** *sphere* | *...red thing left of the blue thing?* **A:** *cube*

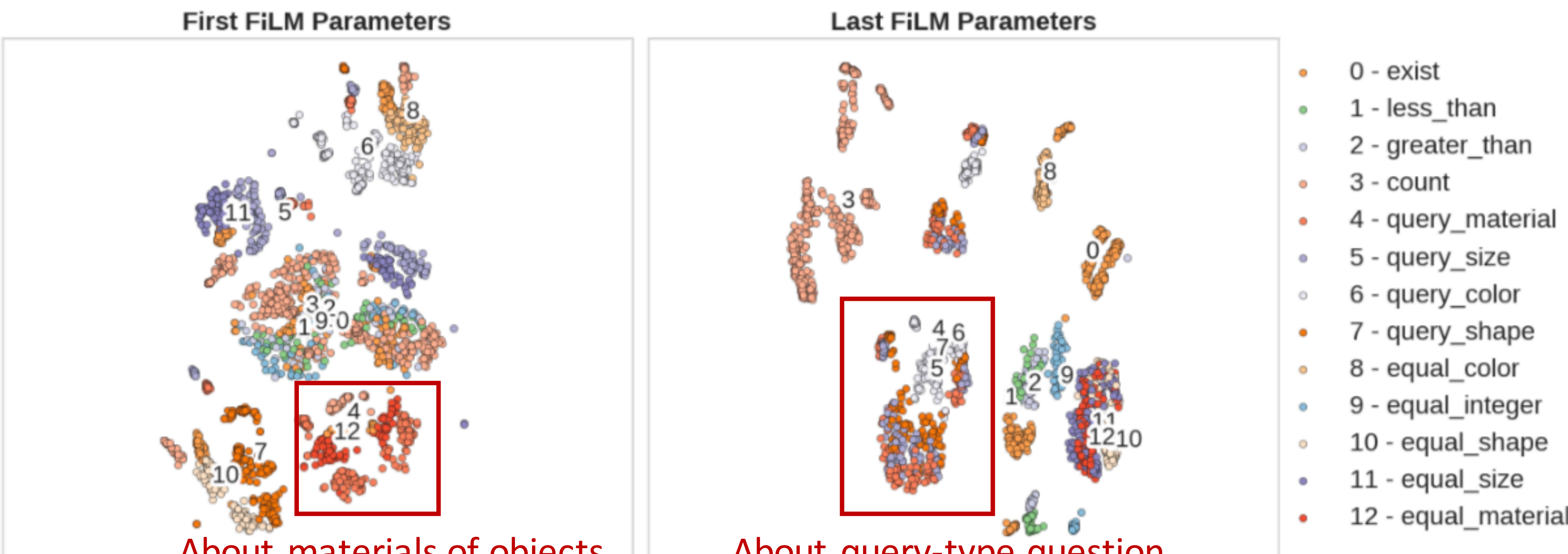
- Performance of FiLM on CLEVR (MODERN paper do not use CLEVR dataset)

Model	**Overall**	Count	Exist	Compare Numbers	Query Attribute	Compare Attribute
Human (Johnson et al. 2017b)	92.6	86.7	96.6	86.5	95.0	96.0
Q-type baseline (Johnson et al. 2017b)	41.8	34.6	50.2	51.0	36.0	51.3
LSTM (Johnson et al. 2017b)	46.8	41.7	61.1	69.8	36.8	51.8
CNN+LSTM (Johnson et al. 2017b)	52.3	43.7	65.2	67.1	49.3	53.0
CNN+LSTM+SA (Santoro et al. 2017)	76.6	64.4	82.7	77.4	82.6	75.4
N2NMN* (Hu et al. 2017)	83.7	68.5	85.7	84.9	90.0	88.7
PG+EE (9K prog.)* (Johnson et al. 2017b)	88.6	79.7	89.7	79.1	92.6	96.0
PG+EE (700K prog.)* (Johnson et al. 2017b)	96.9	92.7	97.1	**98.7**	98.1	98.9
CNN+LSTM+RN†‡ (Santoro et al. 2017)	95.5	90.1	97.8	93.6	97.9	97.1
CNN+GRU+FiLM	**97.7**	**94.3**	99.1	96.8	99.1	99.1
CNN+GRU+FiLM‡	97.6	**94.3**	**99.3**	93.4	**99.3**	**99.3**

*source: https://arxiv.org/pdf/1709.07871.pdf

Modulating Early Visual Processing by Language [de Vries et al., 2017]

- T-SNE plots of (γ, β) of the first/last FiLM layers
  - The FiLM parameters cluster by low-level reasoning in the first layer, and high-level reasoning in the last layer



About materials of objects

About query-type question

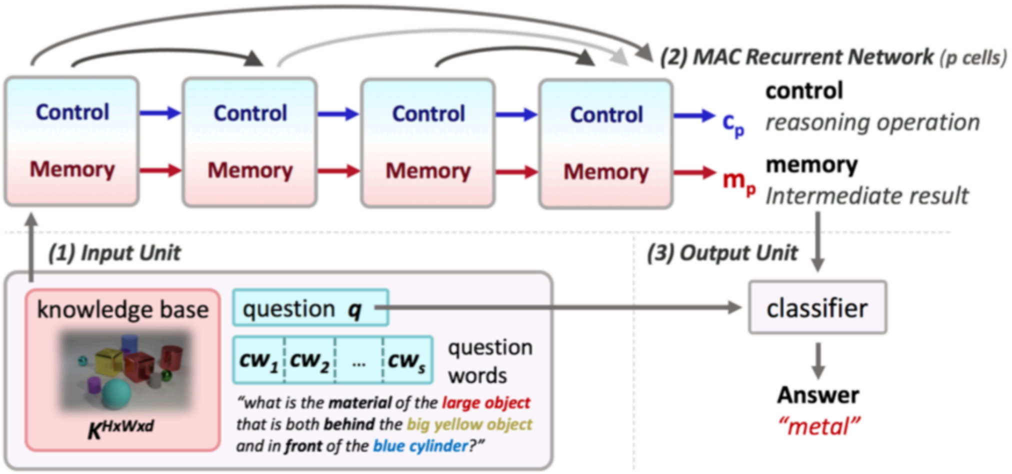
*source: https://arxiv.org/pdf/1709.07871.pdf

Table of Contents

Compositional Attention Networks for Machine Reasoning [Hudson et al., 2018]

- Limitations of previous works
  - Module networks **require strong supervision** about structures, and they are not end-to-end differentiable
  - Augmented CNN approaches do **not have ability for relational reasoning**
    - Relation Network provides only one-step reasoning between objects

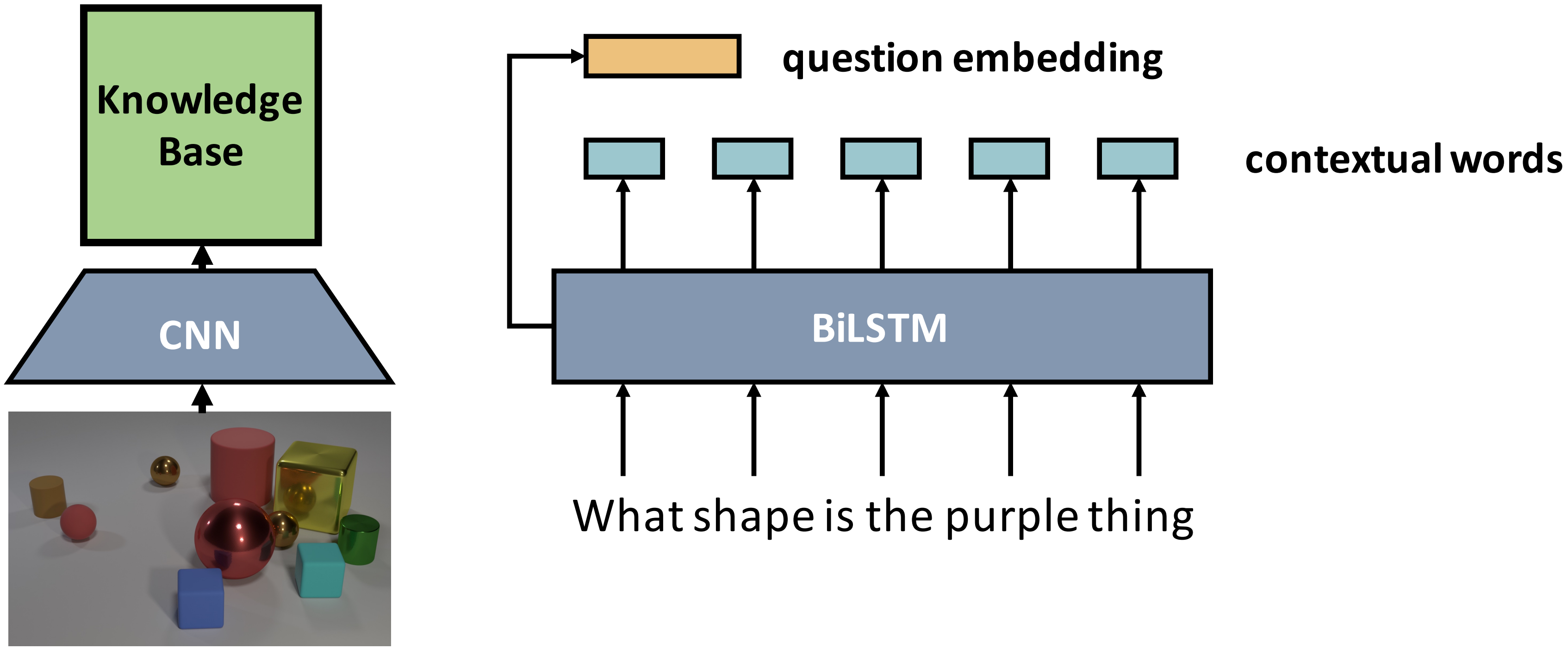
- **Memory-Attention-Composition (MAC) networks**
  - It is **fully differentiable** neural networks
  - It **provides explicit and expressive reasoning** via memory/attention mechanisms



*source: https://arxiv.org/pdf/1803.03067.pdf

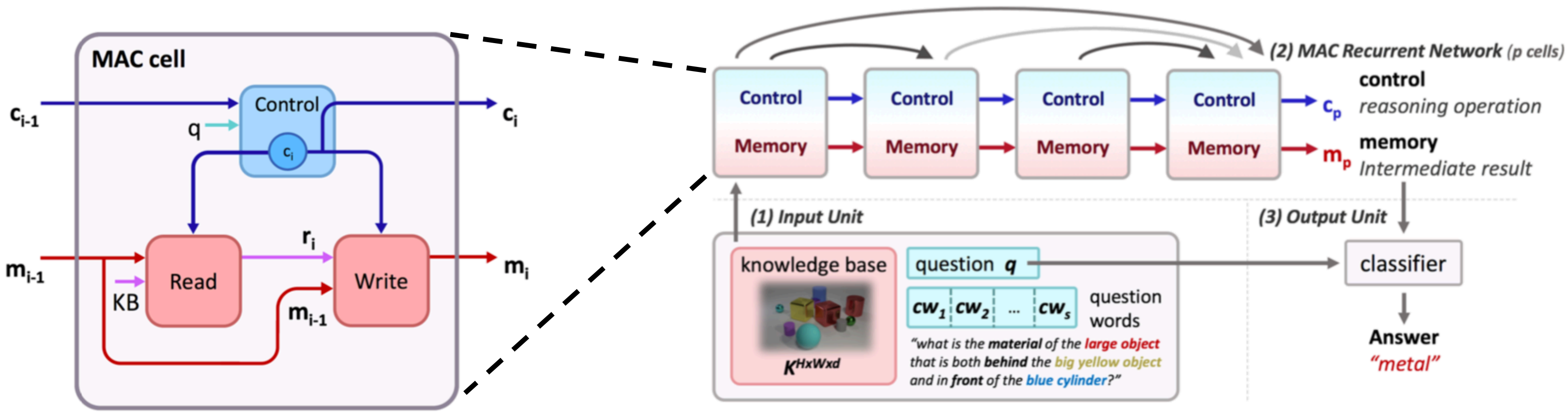
Compositional Attention Networks for Machine Reasoning [Hudson et al., 2018]

- First step (Input Unit)
  - Retrieve **knowledge base** (CNN features) from pre-trained CNN
  - Retrieve **a question embedding** and **contextual word embeddings** using BiLSTM



Compositional Attention Networks for Machine Reasoning [Hudson et al., 2018]

- Main component: **MAC cell**

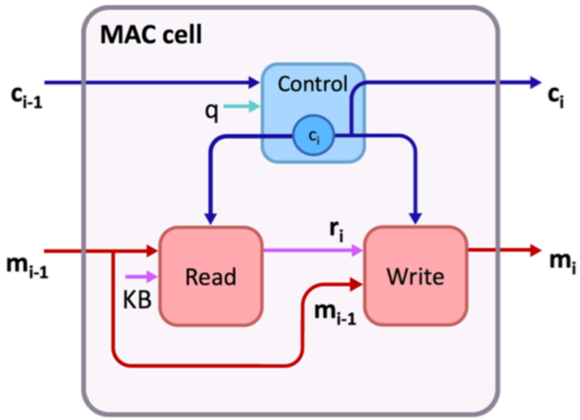


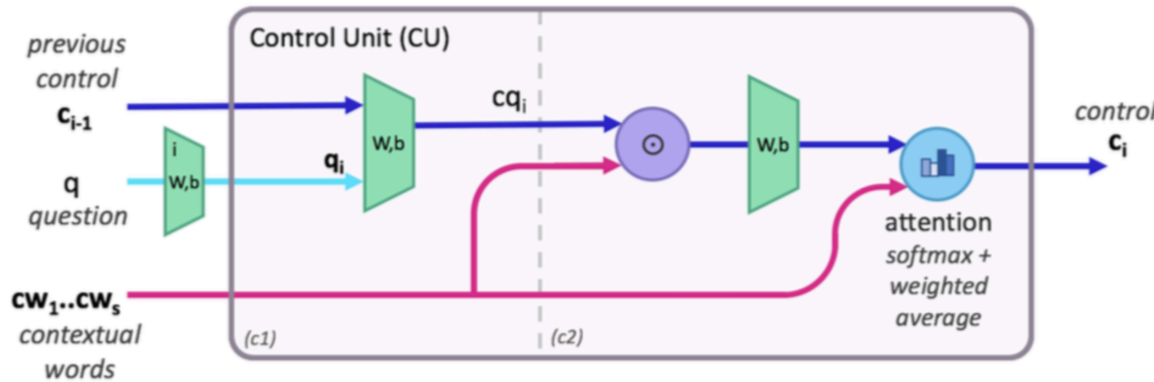
- Each **MAC cell** treats one reasoning step
- Each cell consists of 3 components:
  - **Control Unit** decides which words in question should be focused
  - **Read Unit** retrieves information from knowledge base using control unit
  - **Write Unit** updates memory using retrieved information and control unit
- Multiple **MAC cells** can be recurrently applied
  - i.e., multiple reasoning steps

*source: https://arxiv.org/pdf/1803.03067.pdf

Compositional Attention Networks for Machine Reasoning [Hudson et al., 2018]

- **Control Unit (CU)** in MAC cell

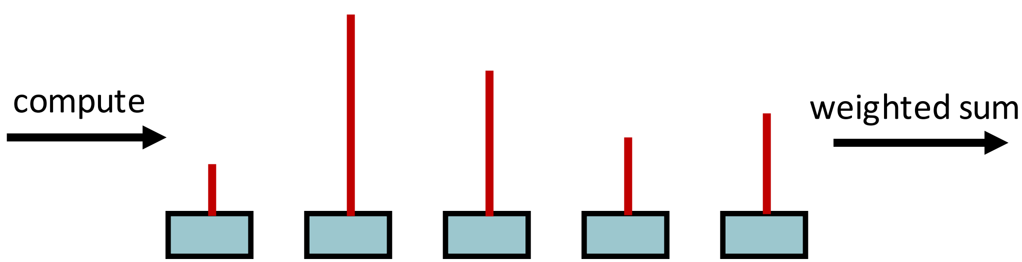




*source: https://arxiv.org/pdf/1803.03067.pdf

- **Control Unit** decides which words in question should be focused
- c_i: control state at step i is weighted sum of contextual word embeddings
  - Compute **attention** using question embedding and previous control state

q: question embedding
c_{i-1}: previous control



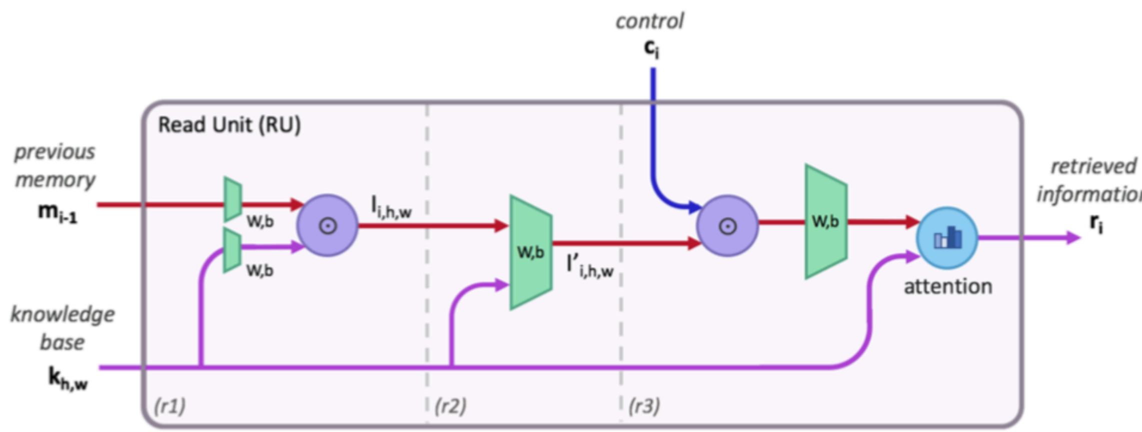
c_i: control

contextual words

Compositional Attention Networks for Machine Reasoning [Hudson et al., 2018]

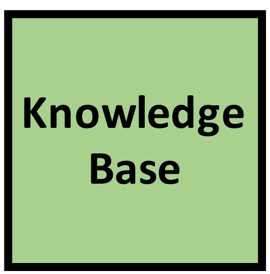
- **Read Unit (RU)** in MAC cell

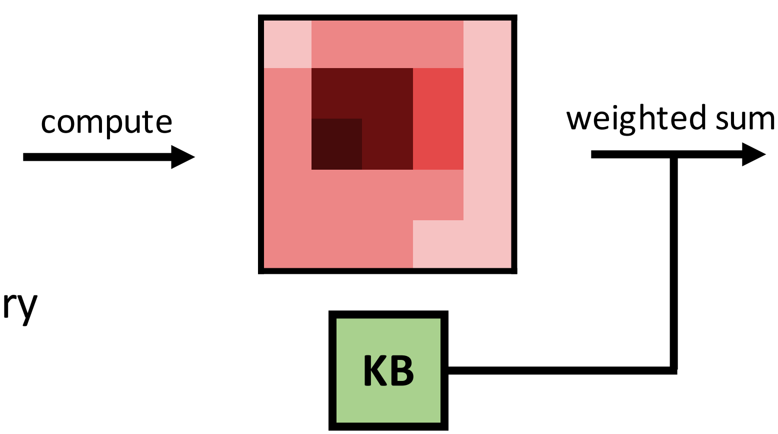
- **Read Unit (RU)** in MAC cell
- **Read Unit** retrieves information from knowledge base using control unit
- m_i: memory state at step i is weighted sum of knowledge base
  - Compute **attention** using current control state and previous memory state



*source: https://arxiv.org/pdf/1803.03067.pdf

- **Read Unit** retrieves information from knowledge base using control unit
- m_i: memory state at step i is weighted sum of knowledge base
  - Compute **attention** using current control state and previous memory state



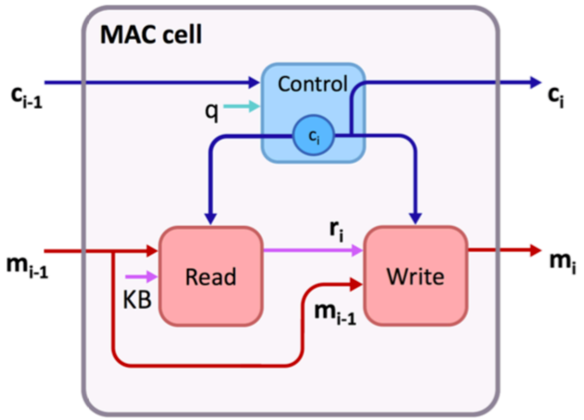


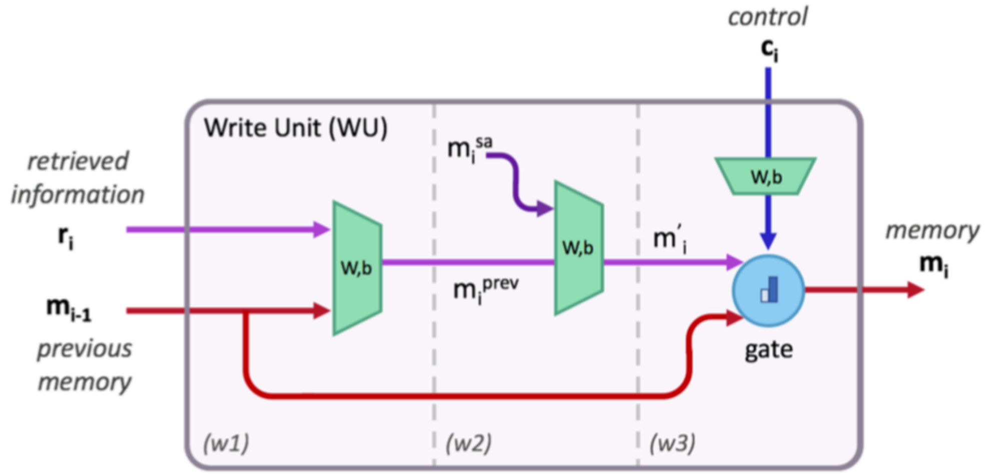
r_i: retrieved information

m_{i-1}: previous memory
c_i: current control

Compositional Attention Networks for Machine Reasoning [Hudson et al., 2018]

- **Write Unit (WU)** in MAC cell



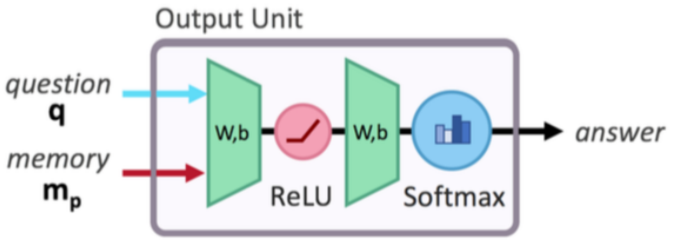


*source: https://arxiv.org/pdf/1803.03067.pdf

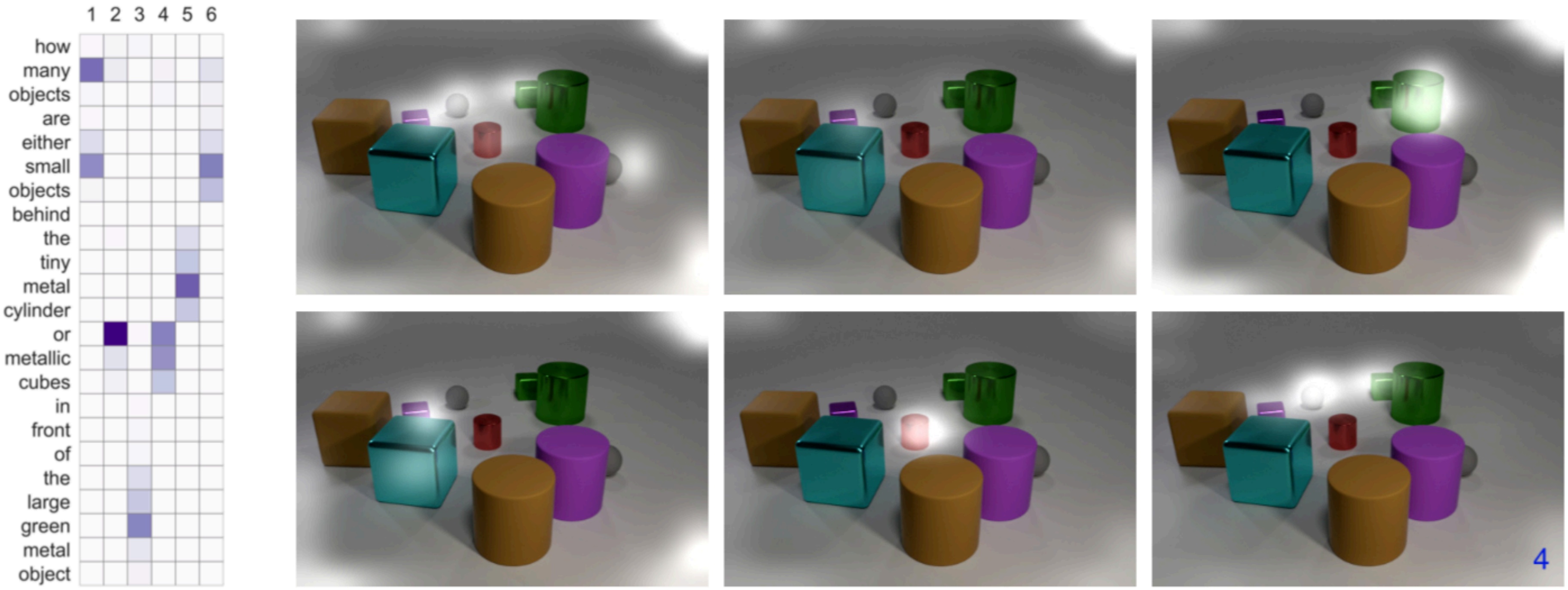
- **Write Unit** updates memory using retrieved information and control unit
- WU uses self-attention, i.e., use previous memory states $m_1, m_2, \dots, m_{i-1}$
  - This provides **non-sequential reasoning processes**
- WU uses memory gate, i.e., $m_i = g m_{i-1} + (1-g) m'_i$
  - Dynamically decide how much information should be updated

Compositional Attention Networks for Machine Reasoning [Hudson et al., 2018]

- Last step (Output Unit)
  - Use the question embedding and last memory state



- **Attention** maps produced by MAC network



*source: https://arxiv.org/pdf/1803.03067.pdf

Compositional Attention Networks for Machine Reasoning [Hudson et al., 2018]

- The state-of-the-art performance on CLEVR dataset

*source: https://arxiv.org/pdf/1803.03067.pdf

Model	CLEVR Overall	Count	Exist	Compare Numbers	Query Attribute	Compare Attribute	Humans before FT	Humans after FT
Human (Johnson et al., 2017b)	92.6	86.7	96.6	86.5	95.0	96.0	-	-
Q-type baseline (Johnson et al., 2017b)	41.8	34.6	50.2	51.0	36.0	51.3	-	-
LSTM (Johnson et al., 2017b)	46.8	41.7	61.1	69.8	36.8	51.8	27.5	36.5
CNN+LSTM (Johnson et al., 2017b)	52.3	43.7	65.2	67.1	49.3	53.0	37.7	43.2
CNN+LSTM+SA+MLP (Johnson et al., 2017a)	73.2	59.7	77.9	75.1	80.9	70.8	50.4	57.6
N2NMN* (Hu et al., 2017)	83.7	68.5	85.7	84.9	90.0	88.7	-	-
PG+EE (9K prog.)* (Johnson et al., 2017b)	88.6	79.7	89.7	79.1	92.6	96.0	-	-
PG+EE (18K prog.)* (Johnson et al., 2017b)	95.4	90.1	97.3	96.5	97.4	98.0	54.0	66.6
PG+EE (700K prog.)* (Johnson et al., 2017b)	96.9	92.7	97.1	98.7	98.1	98.9	-	-
CNN+LSTM+RN†‡ (Santoro et al., 2017)	95.5	90.1	97.8	93.6	97.9	97.1	-	-
CNN+GRU+FiLM (Perez et al., 2017)	97.7	94.3	99.1	96.8	99.1	99.1	56.6	75.9
CNN+GRU+FiLM‡ (Perez et al., 2017)	97.6	94.3	99.3	93.4	99.3	99.3	-	-
MAC	**98.9**	**97.1**	**99.5**	**99.1**	**99.5**	**99.5**	**57.4**	**81.5**

Covered in this lecture

References

- [Girshick et al., 2014] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation.", CVPR 2014.
Link: https://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html
- [Girshick et al., 2015] Girshick, Ross, et al. "Fast r-cnn.", ICCV 2015.
Link: https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Girshick_Fast_R-CNN_ICCV_2015_paper.pdf
- [Ren et al., 2015] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks.", NIPS 2015.
Link: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf
- [He et al., 2017] He, Kaiming, et al. "Mask r-cnn.", ICCV 2017.
Link: https://arxiv.org/pdf/1703.06870.pdf
- [Cai et al., 2018] Cai, Zhaowei, and Nuno Vasconcelos. "Cascade R-CNN: Delving into High Quality Object Detection.", CVPR 2018.
Link: http://openaccess.thecvf.com/content_cvpr_2018/CameraReady/2603.pdf
- [Liu et al., 2018] Liu, Li, et al. "Deep learning for generic object detection: A survey." *arXiv preprint arXiv:1809.02165* (2018).
Link: https://arxiv.org/pdf/1809.02165.pdf
- [Redmon et al., 2016] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection.", CVPR 2016.
Link: https://arxiv.org/pdf/1809.02165.pdf
- [Redmon et al., 2017] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." CVPR 2017.
Link: http://openaccess.thecvf.com/content_cvpr_2017/papers/Redmon_YOLO9000_Better_Faster_CVPR_2017_paper.pdf

References

- [Liu et al., 2016] Liu, Wei, et al. "Ssd: Single shot multibox detector.", ECCV 2016.
  Link: https://arxiv.org/pdf/1512.02325
- [Redmon et al., 2018] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).
  Link: https://pjreddie.com/media/files/papers/YOLOv3.pdf
- [Johnson et al., 2017] Johnson, Justin, et al., "CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning", CVPR 2017.
  Link: https://cs.stanford.edu/people/jcjohns/clevr/, https://arxiv.org/pdf/1612.06890.pdf
- [Johnson et al., 2017] Johnson, Justin, et al., "Inferring and Executing Programs for Visual Reasoning", ICCV 2017.
  Link:
  http://openaccess.thecvf.com/content_ICCV_2017/papers/Johnson_Inferring_and_Executing_ICCV_2017_paper.pdf
- [Santoro et al., 2017] Santoro, Adam, et al., "A simple neural network module for relational reasoning", NIPS 2017.
  Link: https://papers.nips.cc/paper/7082-a-simple-neural-network-module-for-relational-reasoning.pdf
- [de Vries et al., 2017] de Vries, Harm, et al., "Modulating early visual processing by language", NIPS 2017
  Link: https://papers.nips.cc/paper/7237-modulating-early-visual-processing-by-language.pdf
- [Perez et al., 2018] Perez, Ethan, et al., "FiLM: Visual Reasoning with a General Conditioning Layer", AAAI 2018
  Link: https://arxiv.org/abs/1709.07871
- [Hudson et al., 2018] Hudson, Drew A., et al., "Compositional Attention Networks for Machine Reasoning", ICLR 2018
  Link: https://arxiv.org/abs/1803.03067