Vision beyond classification

Advanced models for Computer Vision

“

A picture is worth a thousand words

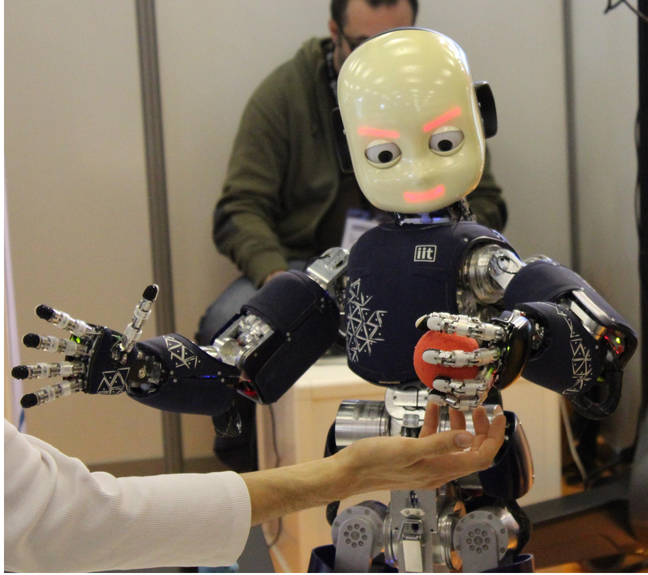
Classification models learn only a few

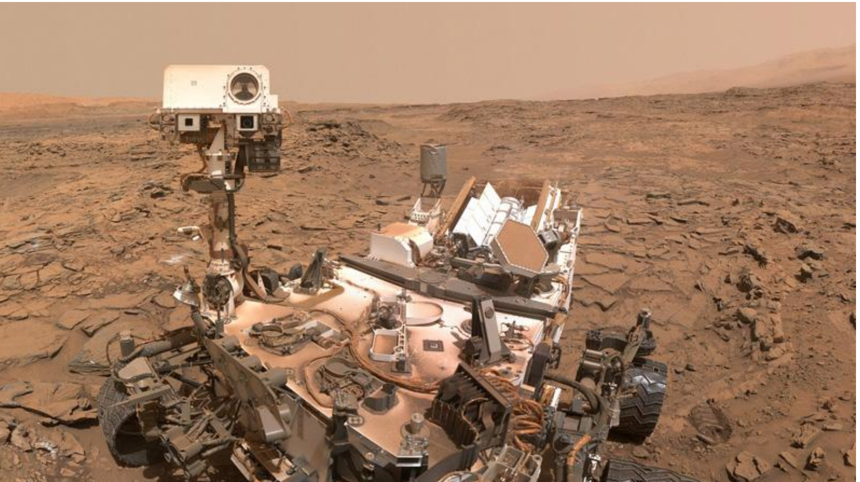
Resnet-50: bicycle, garden

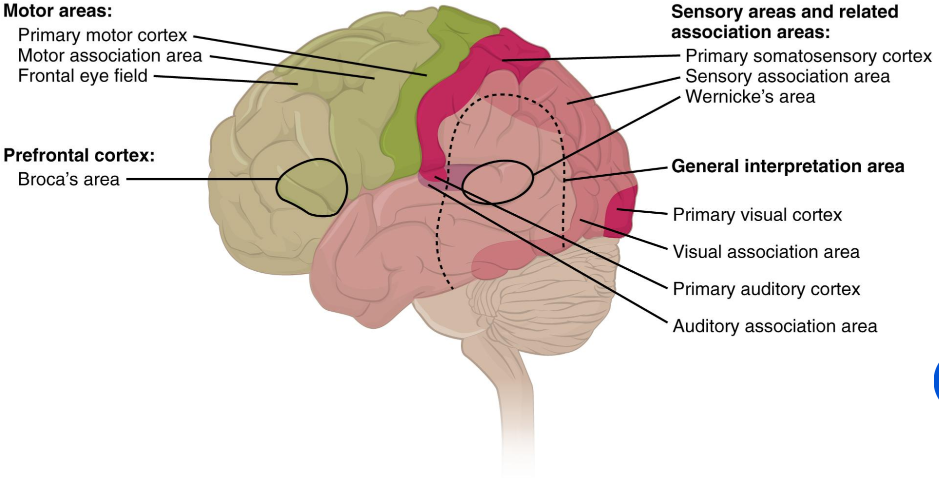
'A Walk On The Bike' By Alexandr Vlasyuk

Holy grail

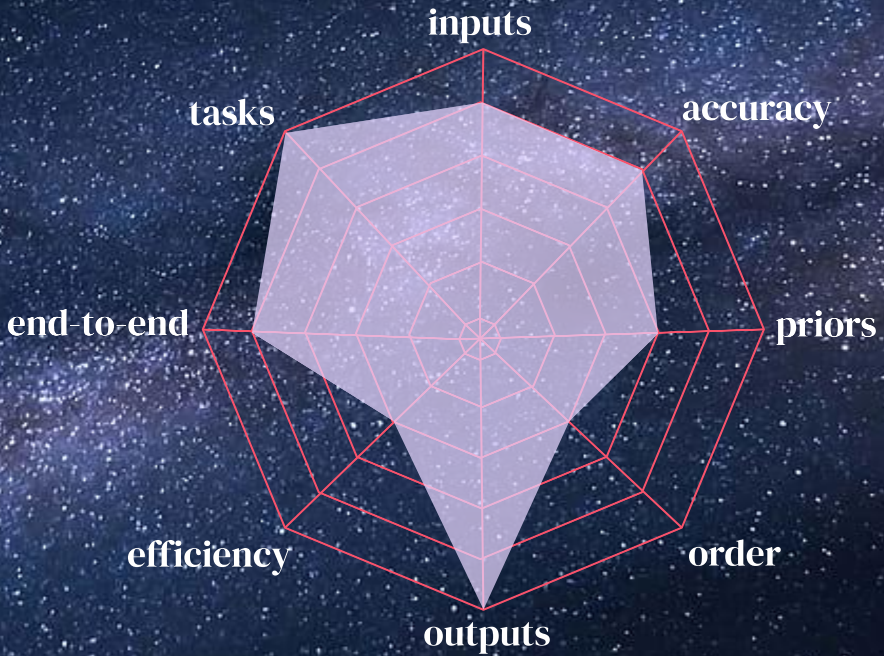
a model that achieves human level scene understanding






Motor areas:
Primary motor cortex
Motor association area
Frontal eye field
Prefrontal cortex:
Broca's area
Sensory areas and related association areas:
Primary somatosensory cortex
Sensory association area
Wernicke's area
General interpretation area
Primary visual cortex
Visual association area
Primary auditory cortex
Auditory association area

Sees-it-all model



Beyond supervised image classification

1
Supervised image ~~classification~~

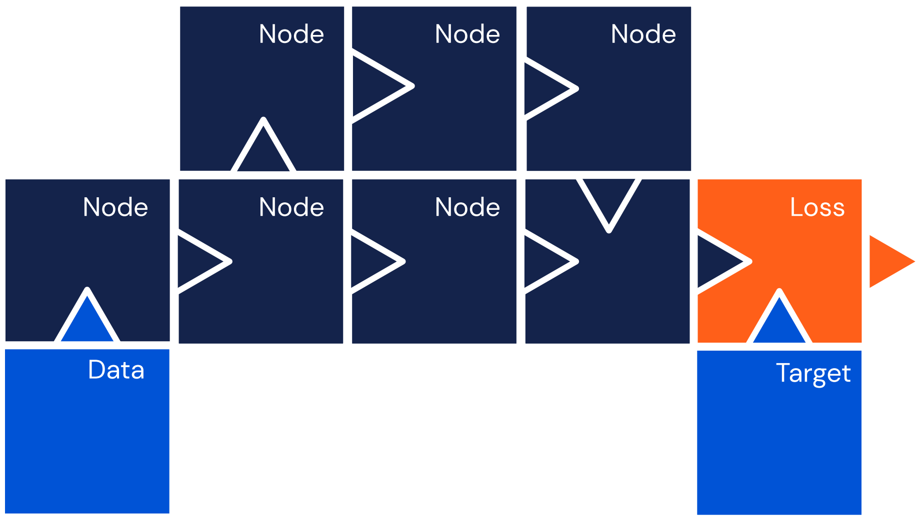
2
Supervised ~~image~~ classification

3
~~Supervised~~ image classification

4
Open questions

The deep learning puzzle

By the end of this lecture, you will know how to redefine these building blocks to perform different visual tasks, using different inputs, and different forms of supervision.



1 Tasks beyond classification

Tasks beyond classification

Task definitions

Object detection

Semantic segmentation

Train and eval

Models and losses

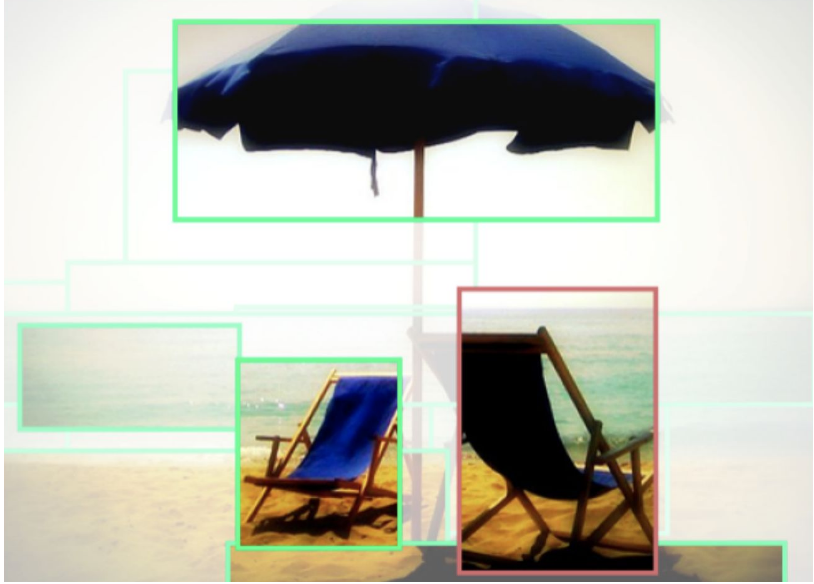
Metrics and benchmarks

Tricks of the trade

Hard negative mining

Transfer learning

Important tasks not covered



Generated Caption: two beach chairs under an umbrella on the beach

Image captioning

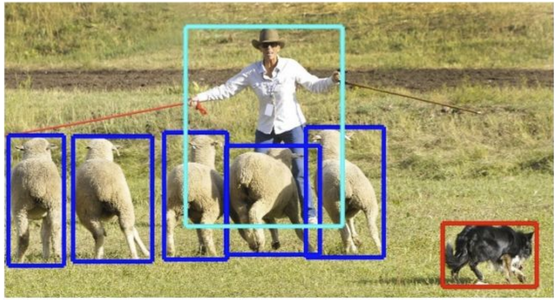
Pose estimation

Image Captioning: Transforming Objects into Words, Herdade et al, 2019
Towards Accurate Multi-person Pose Estimation in the Wild, Papandreou et al, 2017

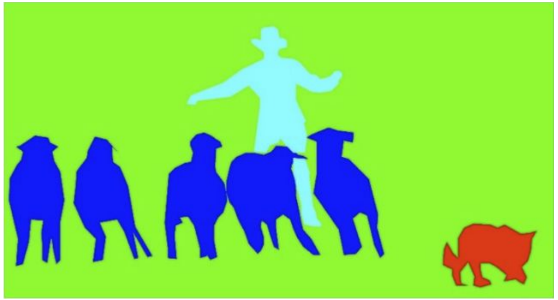
Tasks - Increasing granularity

classification

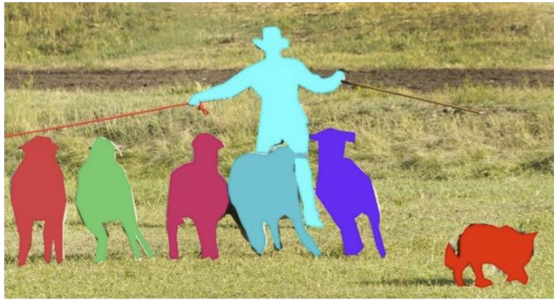
object detection



semantic segmentation



instance segmentation



Figures from *Microsoft COCO: Common Objects in Context, Lin et al, 2014*

Task 1
Object detection

Object detection

Multi-task problem

Classification and localisation

Image from COCO dataset - *Microsoft COCO: Common Objects in Context, Lin et al, 2014*

Object detection

Inputs

- RGB image

Targets

- Class label
- Object bounding box

$$(x_c, y_c, h, w)$$

for all the objects present in the scene

Object detection

Inputs and targets

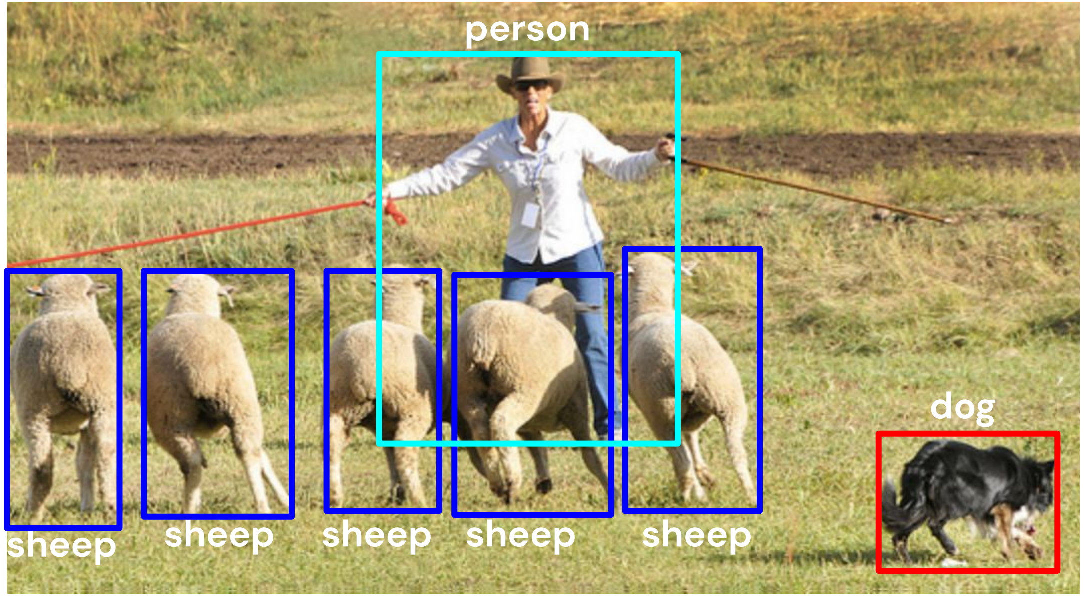


Image from COCO dataset – *Microsoft COCO: Common Objects in Context, Lin et al, 2014*

Object detection

Dataset

$N_{\text{train}}, N_{\text{test}}$ samples

$$
\begin{array}{l}
\{\text{'image'} : p \in [0,1], H \times W \times 3, \\
\quad \text{'objects'} : \\
\quad [\\
\quad \{\text{'label'} : \text{one\_hot}(N), 1 \times N, \\
\quad\quad \text{'bbox'} : (x_c, y_c, h, w) \in \mathbb{R}, 1 \times 4\}, \\
\quad \{\text{'label'} : \text{one\_hot}(N), 1 \times N, \\
\quad\quad \text{'bbox'} : (x_c, y_c, h, w) \in \mathbb{R}, 1 \times 4\}, \\
\quad \vdots \\
\quad]\}
\end{array}
$$

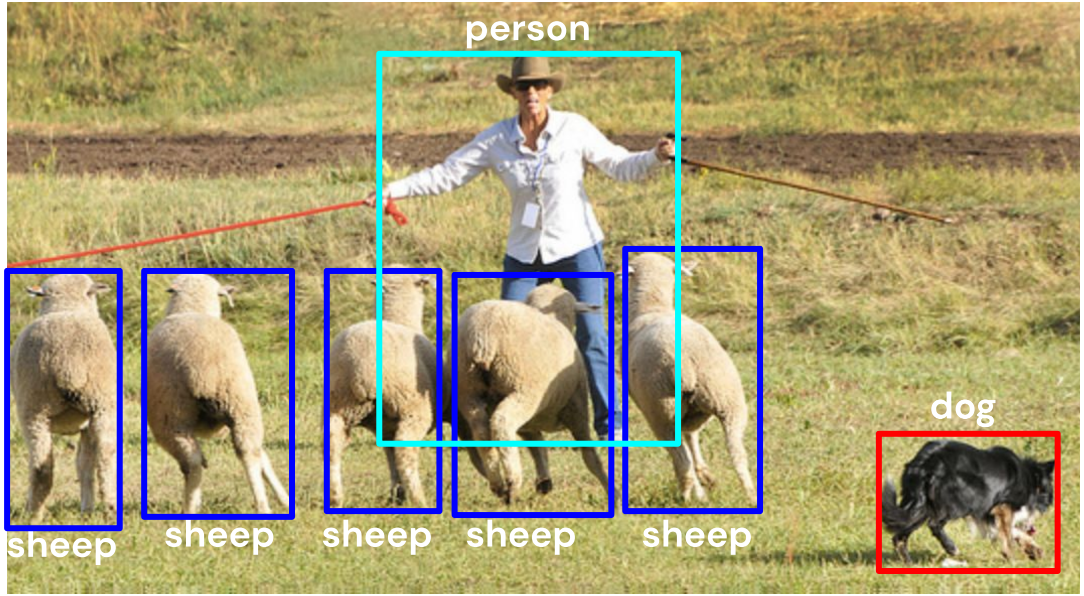


Image from COCO dataset – *Microsoft COCO: Common Objects in Context, Lin et al, 2014*

Object detection

Dataset

$N_{\text{train}}, N_{\text{test}}$ samples

$$\{\text{'image'} : p \in [0,1], H \times W \times 3,$$
$$\text{'objects'} :$$
$$[$$
$$\{\text{'label'} : \text{one\_hot}(N), 1 \times N,$$
$$\text{'bbox'} : (x_c, y_c, h, w) \in \mathbb{R}, 1 \times 4\},$$
$$\{\text{'label'} : \text{one\_hot}(N), 1 \times N,$$
$$\text{'bbox'} : (x_c, y_c, h, w) \in \mathbb{R}, 1 \times 4\},$$
$$\vdots$$
$$]\}$$

How to learn to predict bbox coordinates?

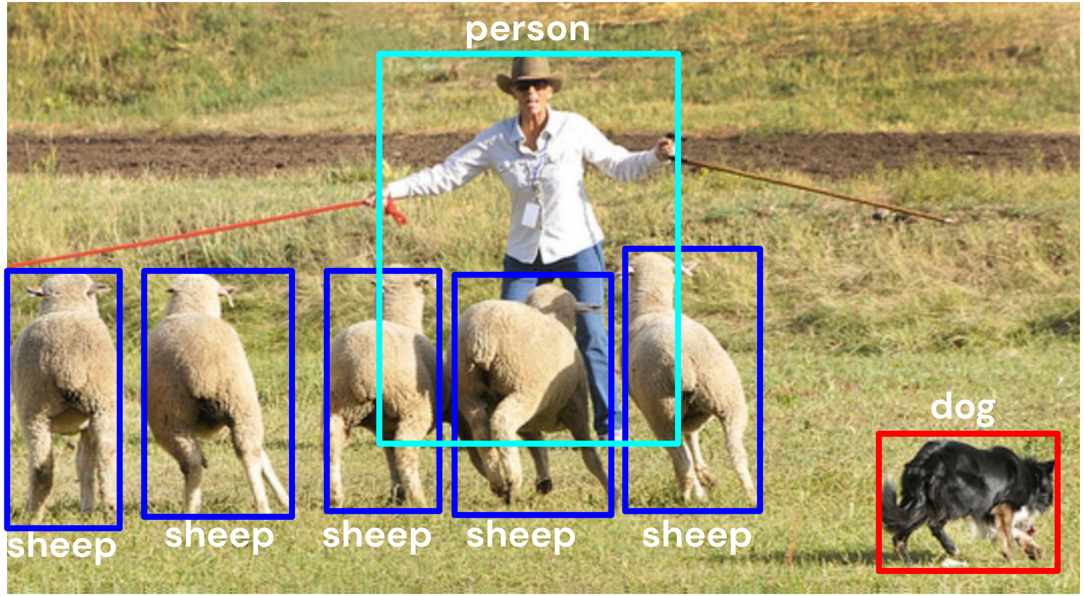
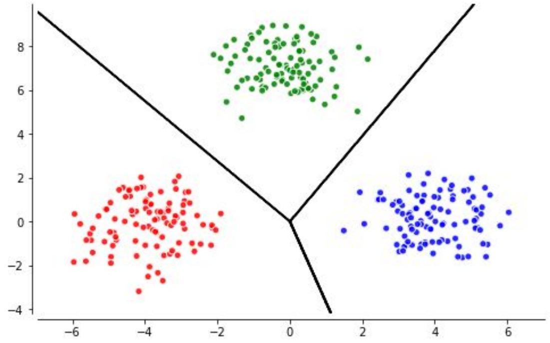


Image from COCO dataset - *Microsoft COCO: Common Objects in Context, Lin et al, 2014*

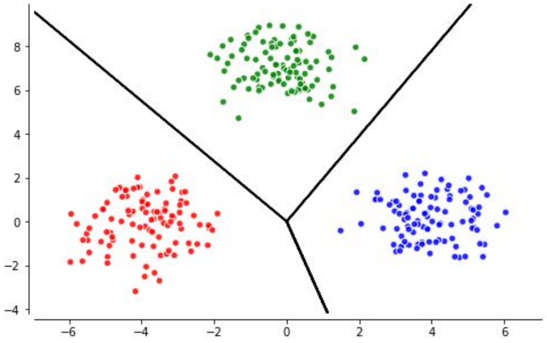
Recap: Softmax + cross entropy for classification



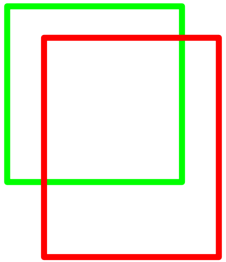
$$\ell_{\mathrm{CE}}(f_{\mathrm{sm}}(\mathbf{x}), \mathbf{t}) = -\sum_{j=1}^{k} \mathbf{t}_j \log[f_{\mathrm{sm}}(\mathbf{x}_j)] = -\sum_{j=1}^{k} \mathbf{t}_j [\mathbf{x}_j - \log \sum_{l=1}^{k} e^{\mathbf{x}_l}]$$

Assign data points to categories; output is discrete.

Bounding box prediction

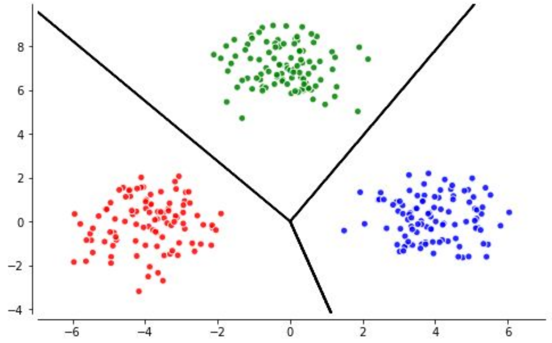


Classification

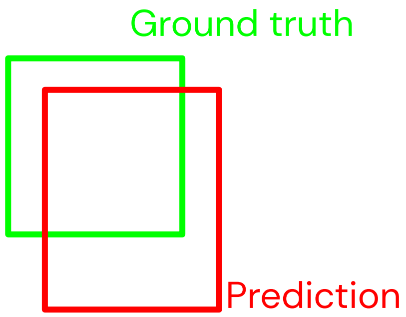


Mistakes are not quantifiable in classification; the data is not *ordered*.

Bounding box prediction



Classification



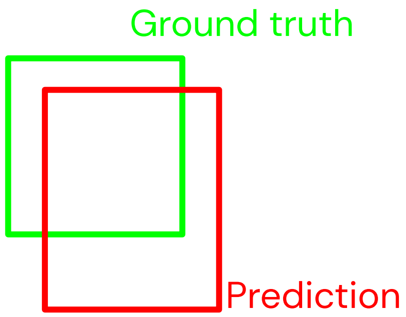
Regression

In classification, the output is discrete*, in regression the output is continuous.

*in classification the output is continuous too, but it represents the probability distribution over the possible classes

Quadratic loss for regression

$$\ell_2(\mathbf{x}, \mathbf{t}) = \|\mathbf{t} - \mathbf{x}\|^2$$



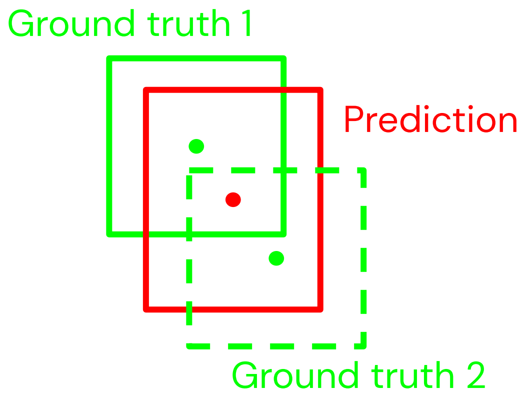
Minimise the mean squared error over samples.

Summary: classification vs regression

Property	Classification	Regression
Basic	map inputs to predefined classes	map inputs to continuous values
Output	discrete values	continuous values
Nature of the data	unordered data	ordered data
Algorithms	logistic regression, decision trees, neural networks	linear regression, neural networks

Quadratic loss for regression

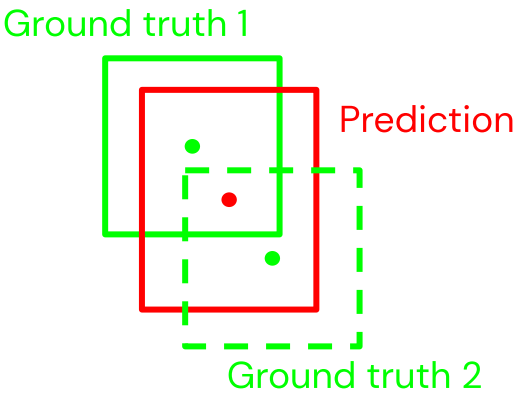
$$\ell_2(\mathbf{x}, \mathbf{t}) = \|\mathbf{t} - \mathbf{x}\|^2$$



How to deal with multiple targets?

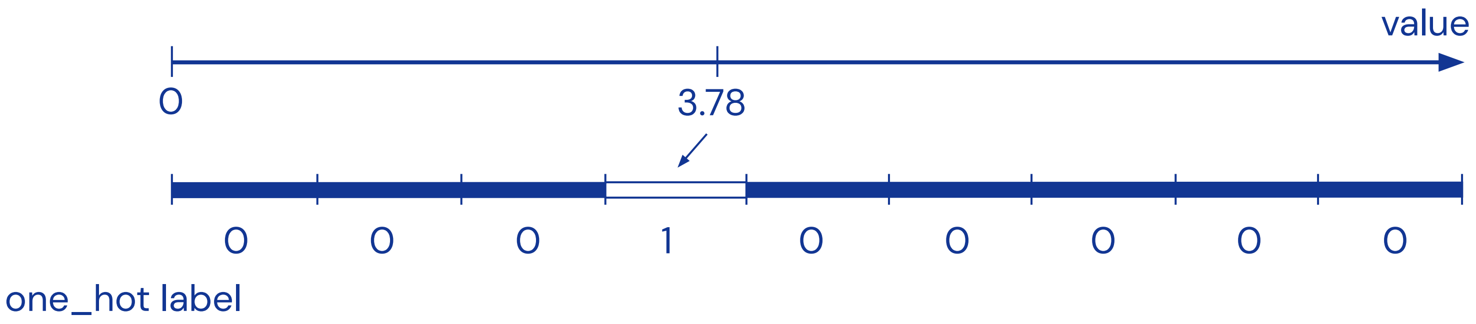
Quadratic loss for regression

$$\ell_2(\mathbf{x}, \mathbf{t}) = \|\mathbf{t} - \mathbf{x}\|^2$$



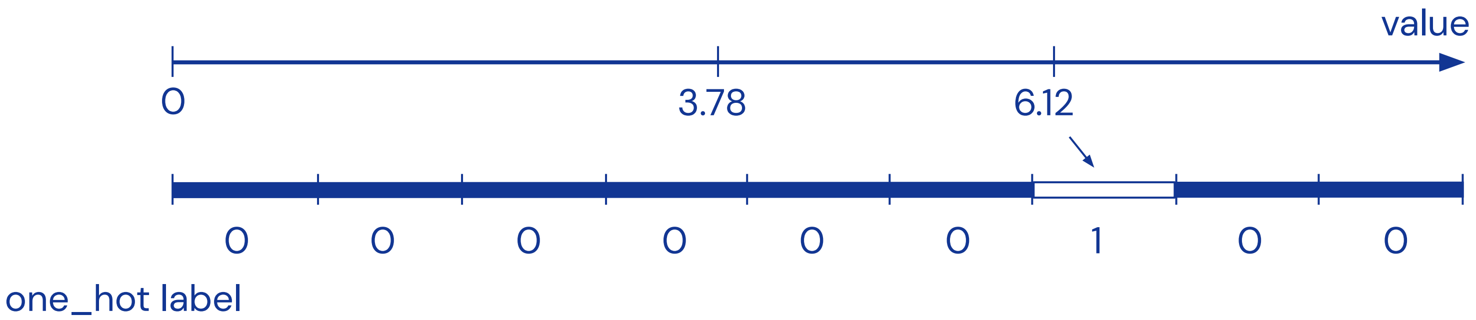
Convert regression into classification, by discretising the output values, and then refine through regression.

Classification then regression



Convert regression into classification, by discretising the output values, and then refine through regression.

Classification then regression



Convert regression into classification, by discretising the output values, and then refine through regression.

Case study 1: Faster R-CNN

Two-stage detector

- Identify good candidate bboxes
- Classify and refine

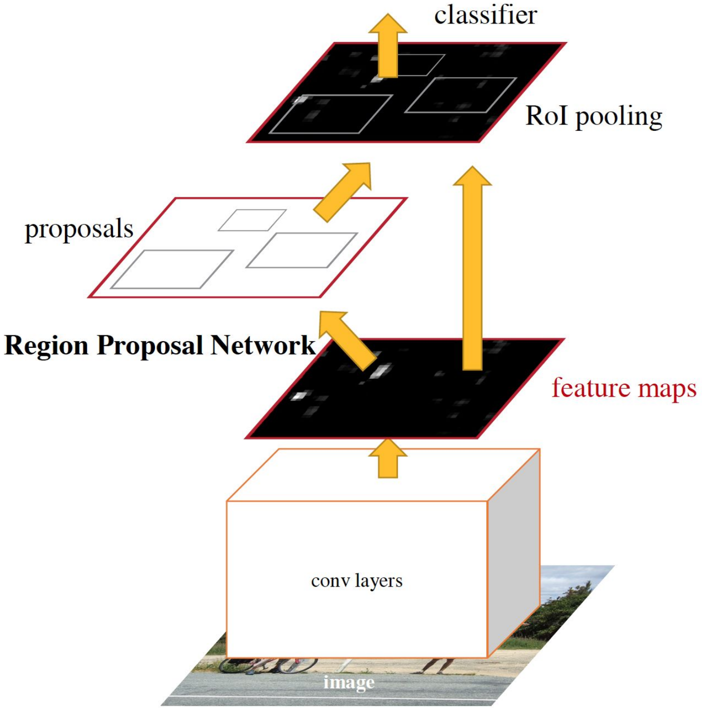


Figure from *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Ren et al, 2016

Case study 1: Faster R-CNN

Identify good candidate bboxes

- Discretise bbox space (x_c, y_c, h, w)
  - anchor points for (x_c, y_c)

Image from COCO dataset – *Microsoft COCO: Common Objects in Context, Lin et al, 2014*

Case study 1: Faster R-CNN

Identify good candidate bboxes

- Discretise bbox space (x_c, y_c, h, w)
  - anchor points for (x_c, y_c)
  - scales and ratios for (h, w)

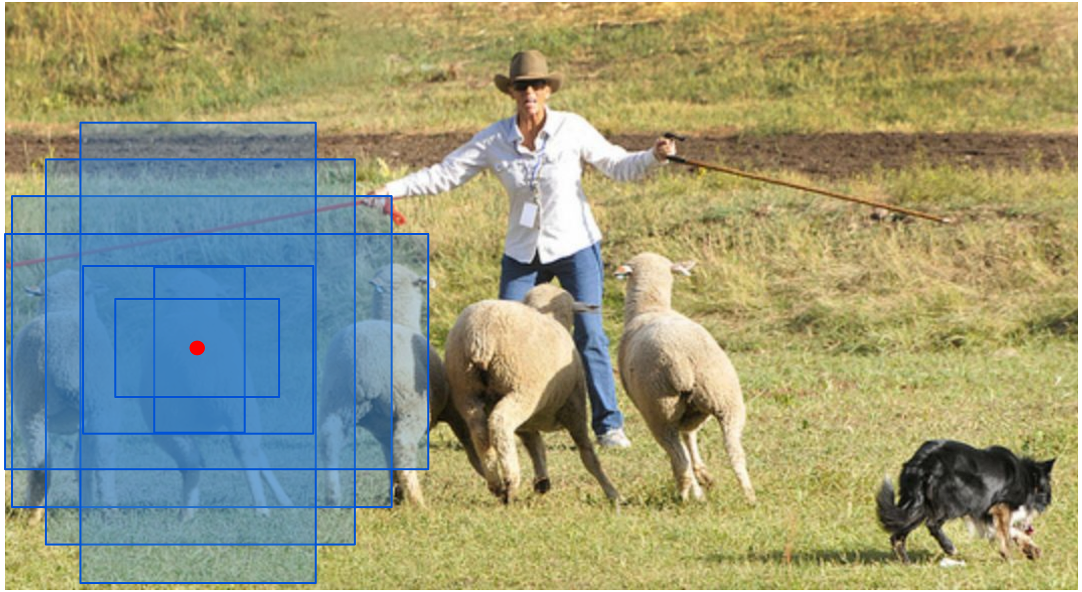


Image from COCO dataset - *Microsoft COCO: Common Objects in Context, Lin et al, 2014*

Case study 1: Faster R-CNN

Identify good candidate bboxes

- Discretise bbox space (x_c, y_c, h, w)
  - anchor points for (x_c, y_c)
  - scales and ratios for (h, w)
- n candidates per anchor
- predict objectness score for each bbox
- sort and keep top K

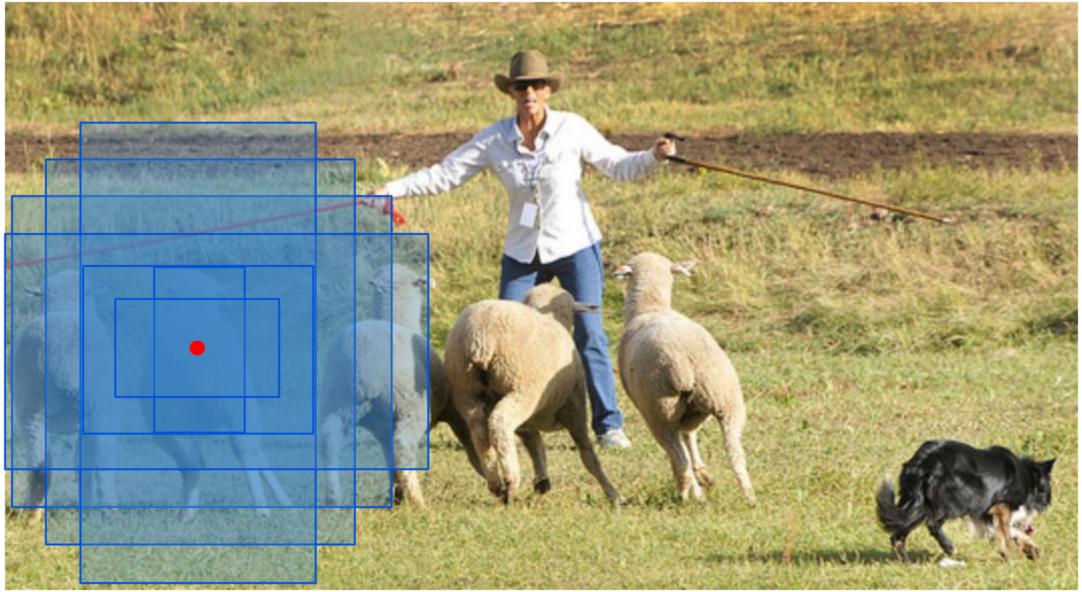


Image from COCO dataset – *Microsoft COCO: Common Objects in Context, Lin et al, 2014*

Case study 1: Faster R-CNN

Identify good candidate bboxes

- Discretise bbox space (x_c, y_c, h, w)
  - anchor points for (x_c, y_c)
  - scales and ratios for (h, w)
- n candidates per anchor
- predict objectness score for each bbox
- sort and keep top K

Refine through regression MLP(4)

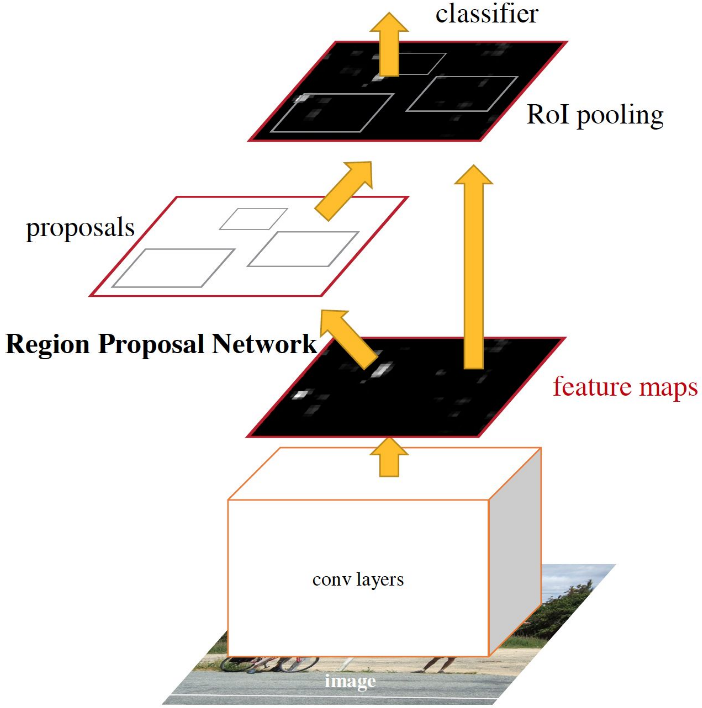


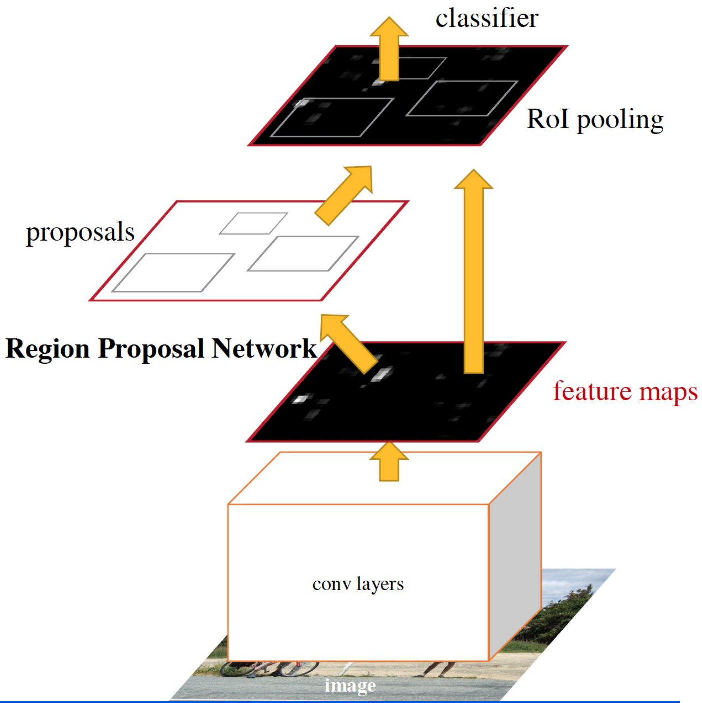
Figure from *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Ren et al, 2016

Case study 1: Faster R-CNN

Identify good candidate bboxes

- Discretise bbox space (x_c, y_c, h, w)
  - anchor points for (x_c, y_c)
  - scales and ratios for (h, w)
- n candidates per anchor
- predict objectness score for each bbox
- sort and keep top K

Refine through regression MLP(4)



Good accuracy @ ~5fps speed

Figure from *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Ren et al, 2016

Case study 1: Faster R-CNN

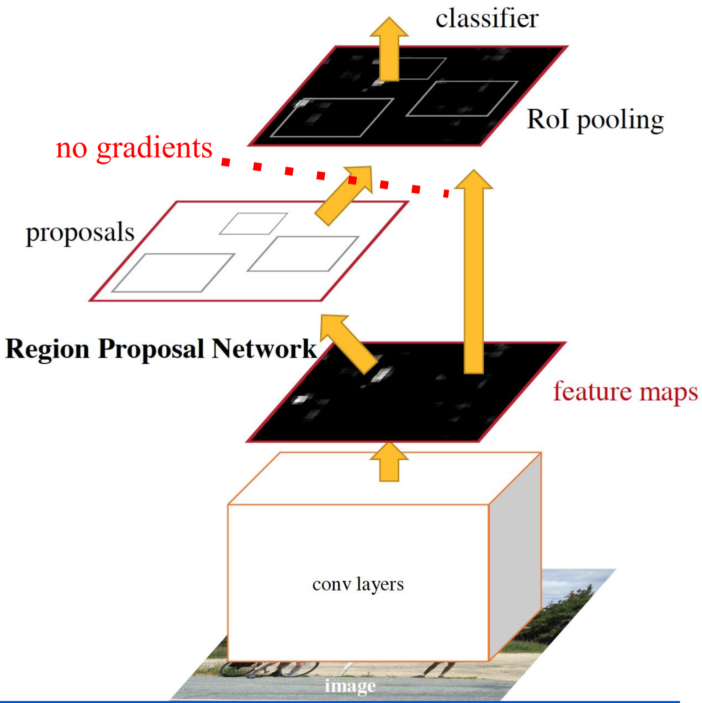
Want to learn more?

Jaderberg et al **Spatial Transformer Networks (2015)**

Identify good candidate bboxes

- Discretise bbox space (x_c, y_c, h, w)
  - anchor points for (x_c, y_c)
  - scales and ratios for (h, w)
- n candidates per anchor
- predict objectness score for each bbox
- sort and keep top K

Refine through regression MLP(4)



Good accuracy @ ~5fps speed

Figure from *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Ren et al, 2016

Case study 2: RetinaNet - one-stage detector

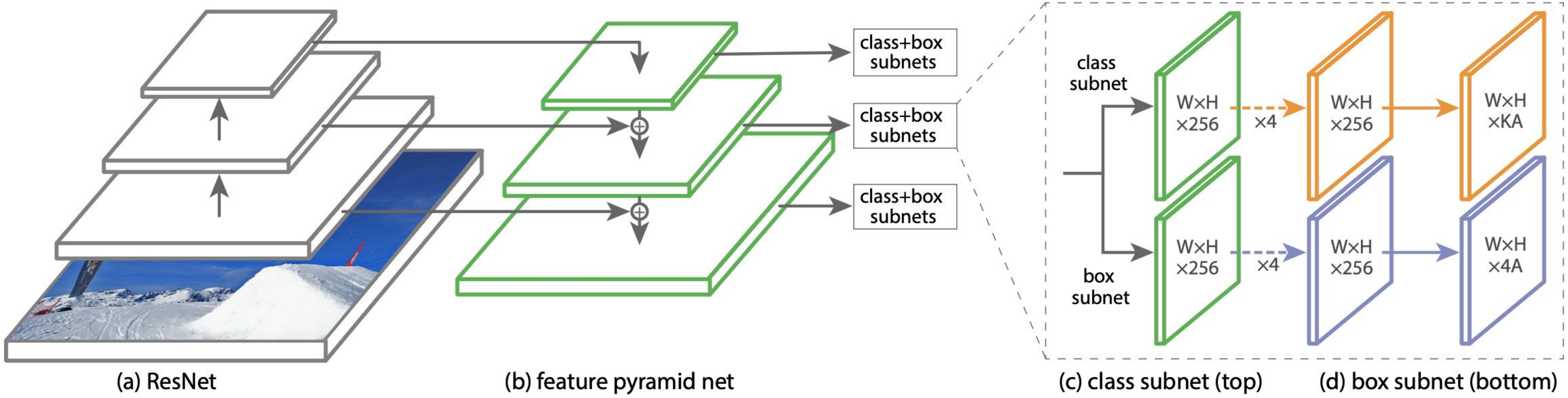
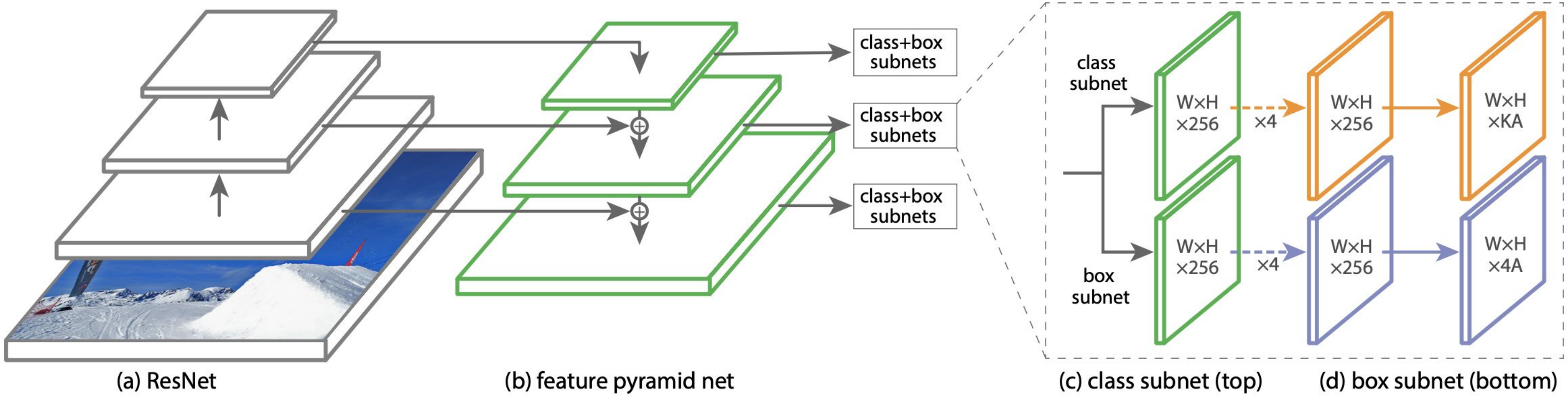


Figure from *Focal Loss for Dense Object Detection*, Lin et al, 2017

Case study 2: RetinaNet - one-stage detector



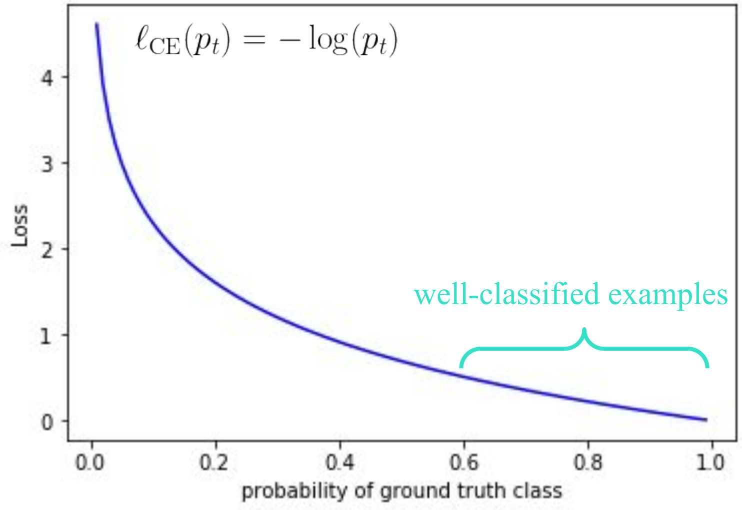
Most of the candidate bboxes are easy negatives: poor learning signal.

Figure from *Focal Loss for Dense Object Detection*, Lin et al, 2017

Issue with one-stage detectors

Most of the candidate bboxes are background, easy to classify.

The accumulated loss of the many easy examples overwhelms the loss of rare useful examples $\ell_{CE}(p > .5) > 0$



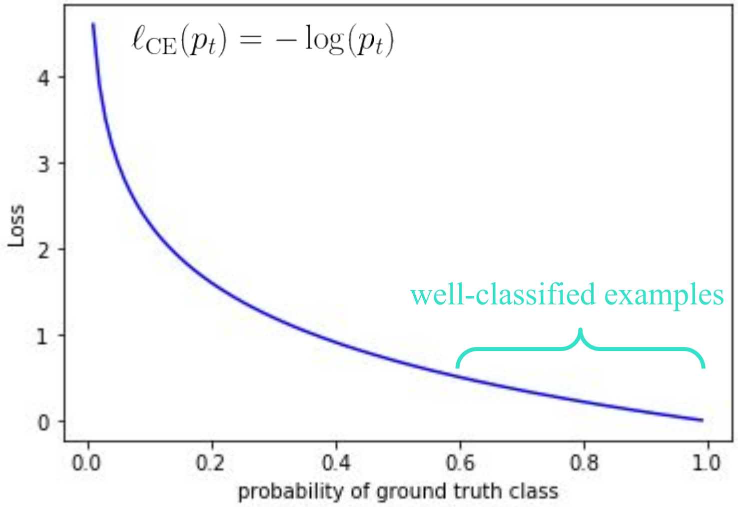
Issue with one-stage detectors

Most of the candidate bboxes are background, easy to classify.

The accumulated loss of the many easy examples overwhelms the loss of rare useful examples $\ell_{CE}(p > .5) > 0$

Faster R-CNN prunes these in stage 1.

One-stage detectors employ **hard negative mining** heuristics.



Hard negative mining

Most of the candidate bboxes are background, easy to classify.

The accumulated loss of the many easy examples overwhelms the loss of rare useful examples $\ell_{CE}(p > .5) > 0$

Faster R-CNN prunes these in stage 1.

One-stage detectors employ **hard negative mining** heuristics.

Hard negative mining for person detector

1. Get set of positive examples
2. Get a random subset of negative examples (full set is too big)
3. Train detector
4. Test on unseen images
5. Identify false positive examples (a person was detected where there was none) = **hard negatives;** add them to the training set
6. Repeat from 3.

Hard negative mining

Most of the candidate bboxes are background, easy to classify.

The accumulated loss of the many easy examples overwhelms the loss of rare useful examples $\ell_{CE}(p > .5) > 0$

Faster R-CNN prunes these in stage 1.

One-stage detectors employ **hard negative mining** heuristics.

Want to learn more?

Sung and Poggio **Learning and Example Selection for Object and Pattern Detection (1994)**

Hard negative mining for person detector

1. Get set of positive examples
2. Get a random subset of negative examples (full set is too big)
3. Train detector
4. Test on unseen images
5. Identify false positive examples (a person was detected where there was none) = **hard negatives;** add them to the training set
6. Repeat from 3.

RetinaNet solution

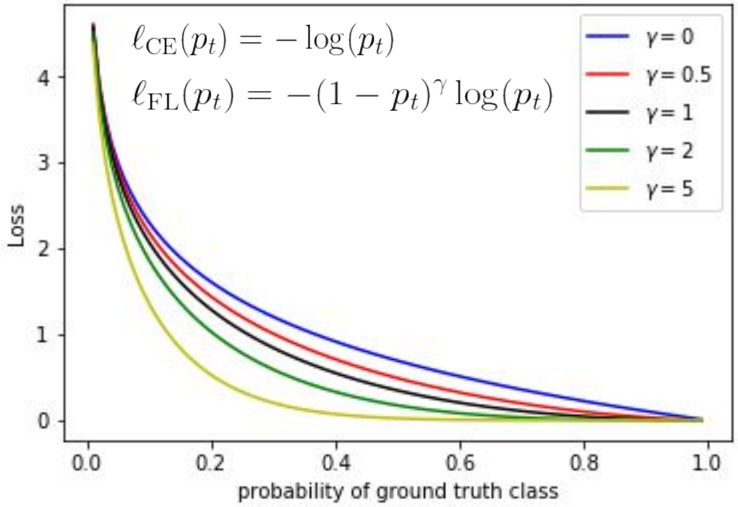
Most of the candidate bboxes are background, easy to classify.

The accumulated loss of the many easy examples overwhelms the loss of rare useful examples $\ell_{CE}(p > .5) > 0$

Faster R-CNN prunes these in stage 1.

One-stage detectors employ **hard negative mining** heuristics.

RetinaNet uses Focal Loss (FL).



RetinaNet solution

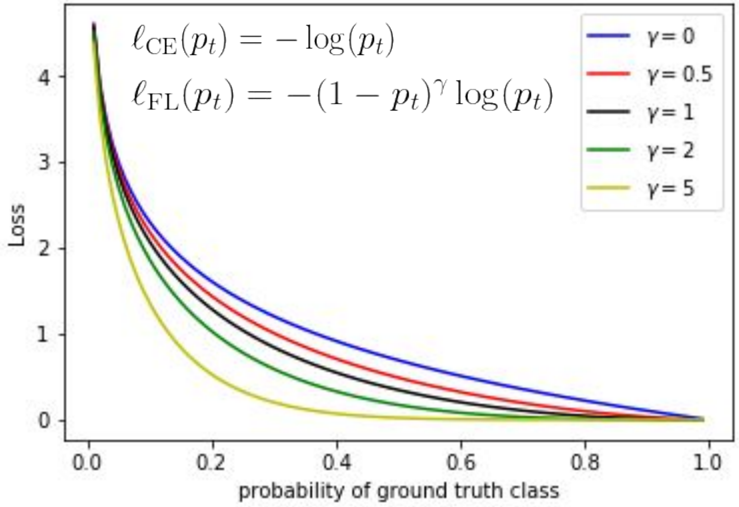
Most of the candidate bboxes are background, easy to classify.

The accumulated loss of the many easy examples overwhelms the loss of rare useful examples $\ell_{CE}(p > .5) > 0$

Faster R-CNN prunes these in stage 1.

One-stage detectors employ **hard negative mining** heuristics.

RetinaNet uses Focal Loss (FL).



Good accuracy @ ~8fps speed

Task 2
Semantic segmentation

Semantic segmentation

Bounding boxes are not good representations for certain types of objects.

We need more refined representations.

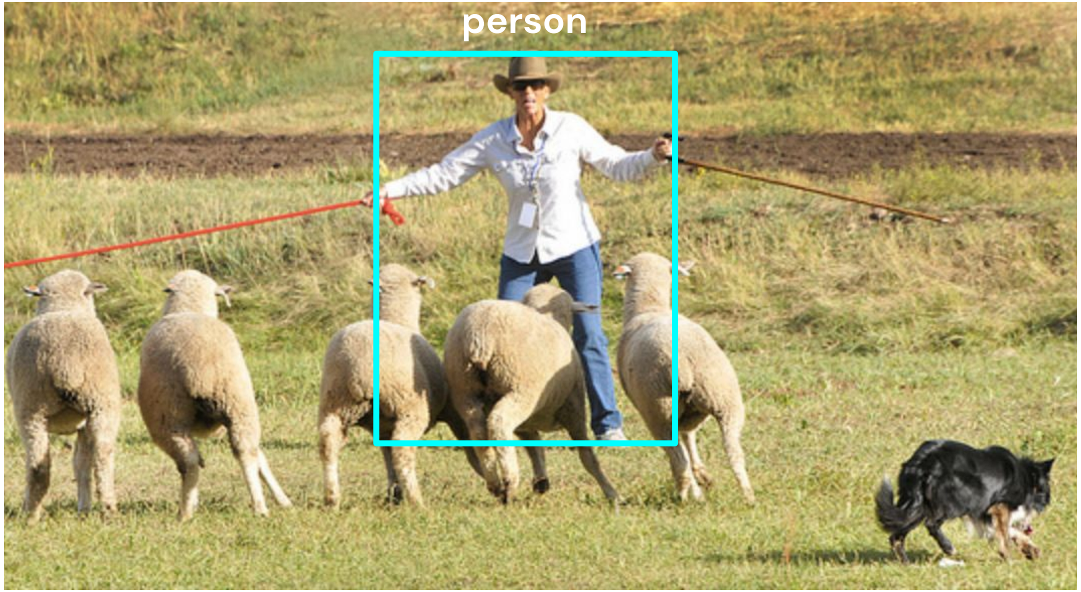


Image from COCO dataset - *Microsoft COCO: Common Objects in Context, Lin et al, 2014*

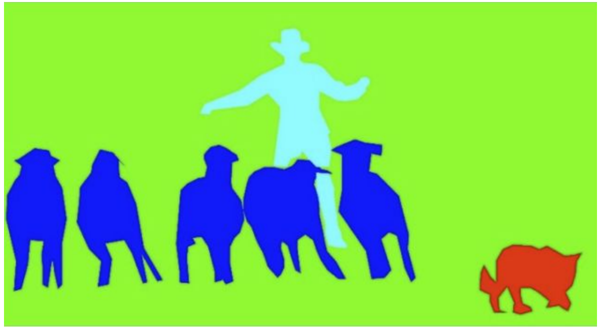
Semantic segmentation

Inputs

- RGB image

Targets

- Class label for every pixel



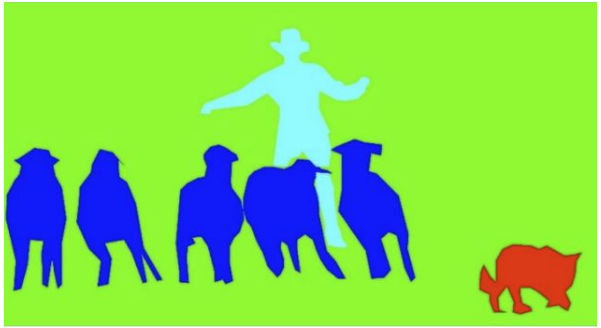
Semantic segmentation

Inputs

- RGB image

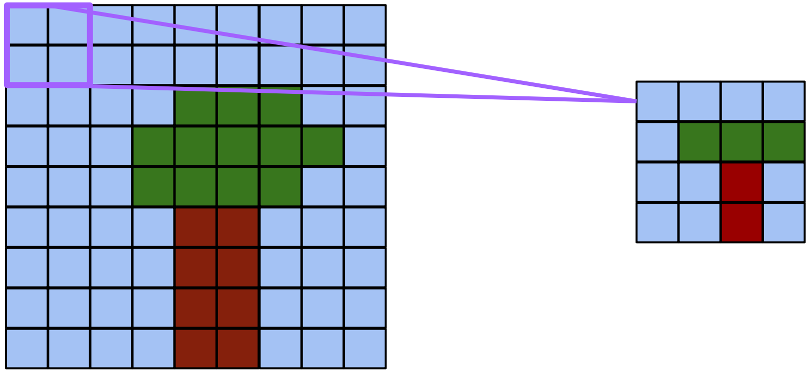
Targets

- Class label for every pixel



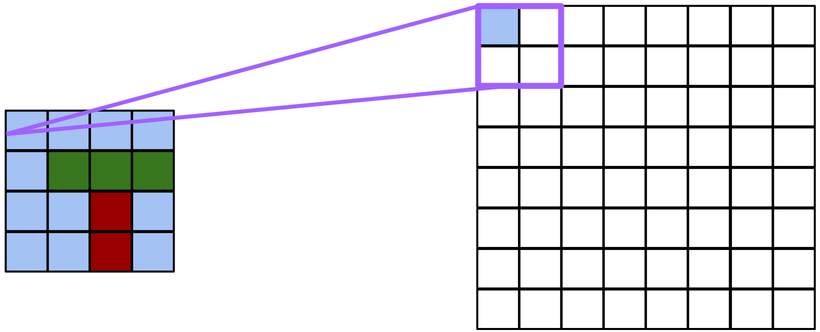
Dense prediction problem – how to generate an output at the same resolution as the input?

Recap: Pooling



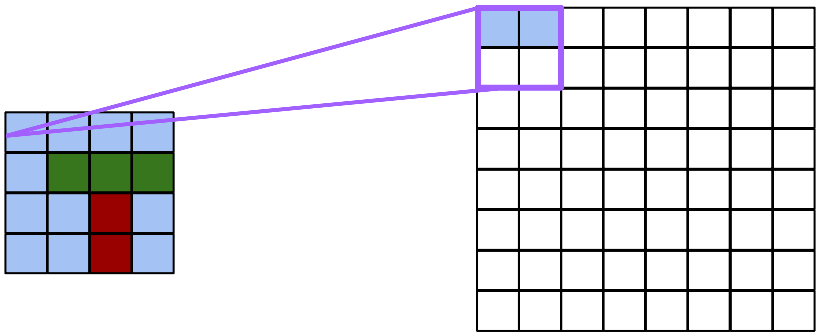
Pooling: compute mean or max over small windows to reduce resolution.

Unpooling



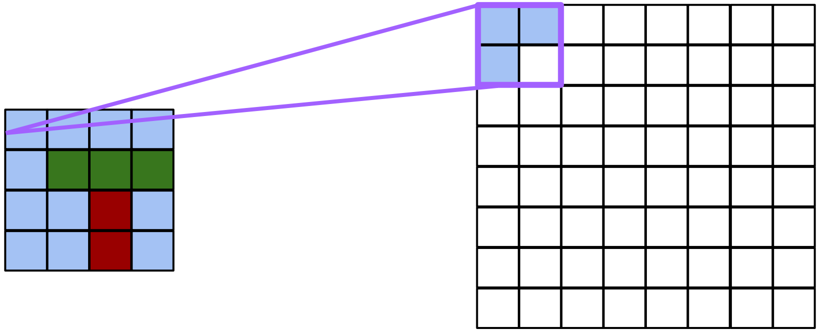
Unpooling: upsample to increase resolution; here 2x2 kernel.

Unpooling



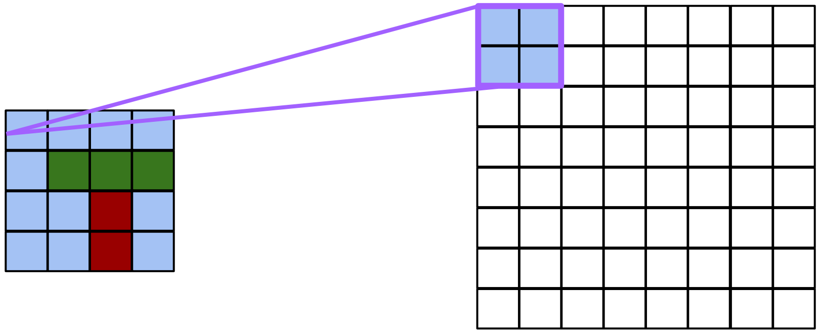
Unpooling: upsample to increase resolution; here 2x2 kernel.

Unpooling



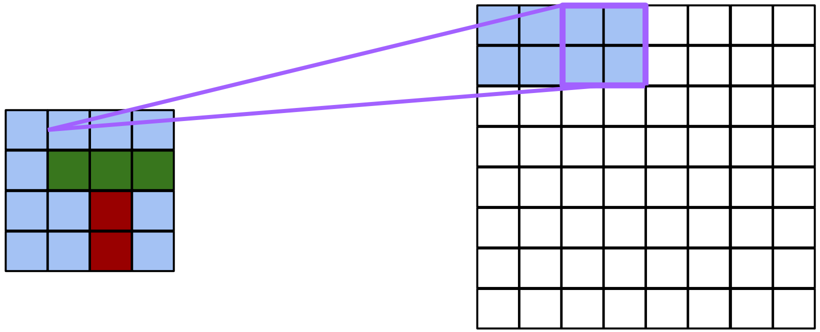
Unpooling: upsample to increase resolution; here 2x2 kernel.

Unpooling



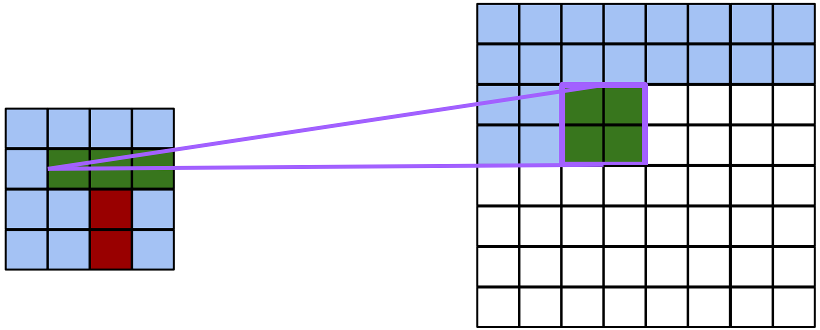
Unpooling: upsample to increase resolution; here 2x2 kernel.

Unpooling



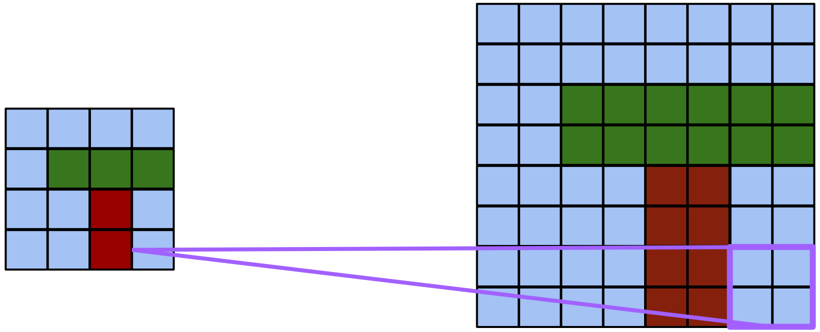
Unpooling: upsample to increase resolution; here 2x2 kernel.

Unpooling



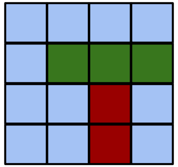
Unpooling: upsample to increase resolution; here 2x2 kernel.

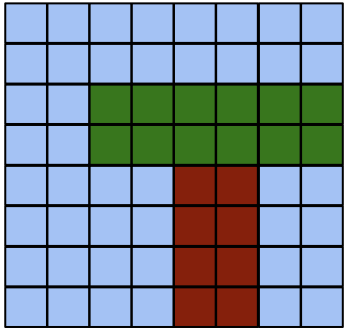
Unpooling



Unpooling: upsample to increase resolution; here 2x2 kernel.

Unpooling





Other upsampling methods

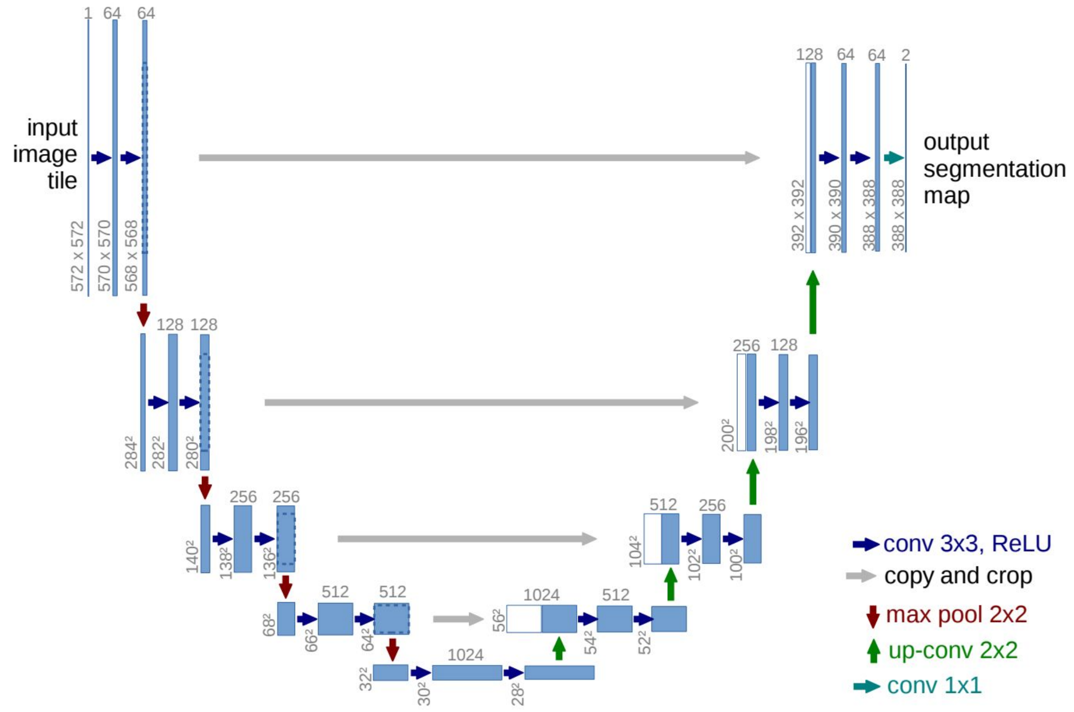
- unpooling with indices *SegNet*
- Deconvolutions *DeconvNet*

Unpooling: upsample to increase resolution; here 2x2 kernel.

SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, Badrinarayanan et al, 2016
DeconvNet: *Learning Deconvolution Network for Semantic Segmentation*, Noh et al, 2015

Case study: U-NET

- Encoder – decoder model
- Skip connections to preserve details



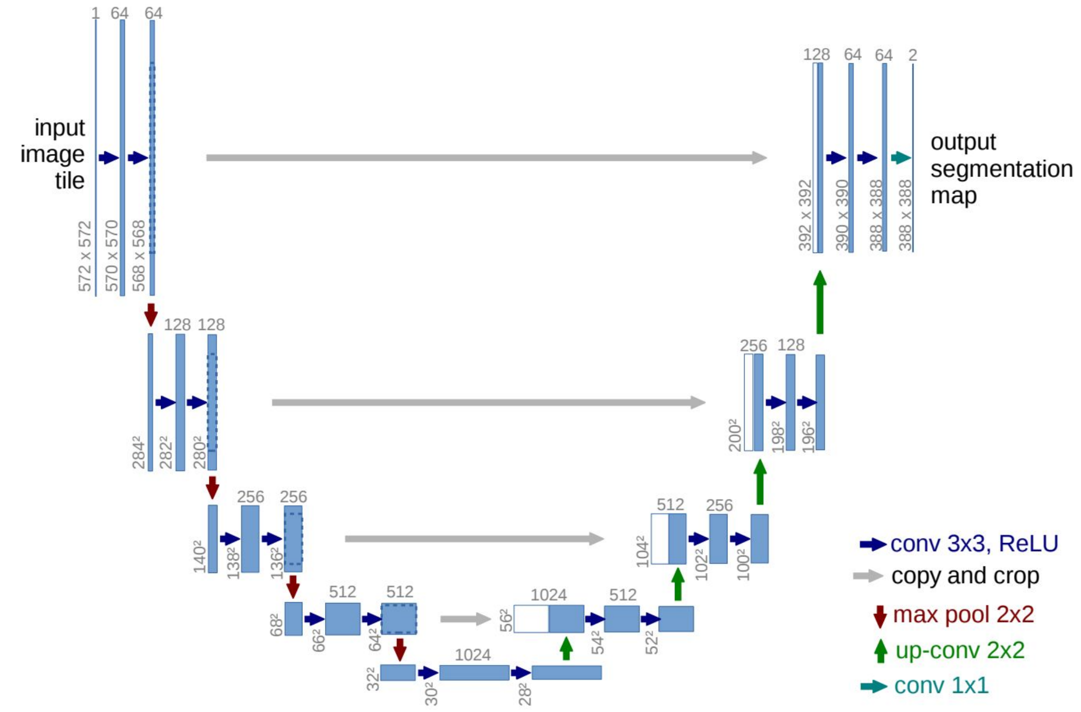
U-Net: Convolutional Networks for Biomedical Image Segmentation, Ronneberger et al, 2015

Case study: U-NET

- Output $H \times W \times N_{\text{classes}}$

- Loss: pixel-wise cross entropy

$$\ell_{\text{CE}}(\mathbf{p}, \mathbf{t}) = -\frac{1}{HW} \sum_{i=1}^{HW} \sum_{j=1}^{N_{\text{classes}}} \mathbf{t}_{ij} \log \mathbf{p}_{ij}$$



U-Net: Convolutional Networks for Biomedical Image Segmentation, Ronneberger et al, 2015

Recall RetinaNet - same U shape

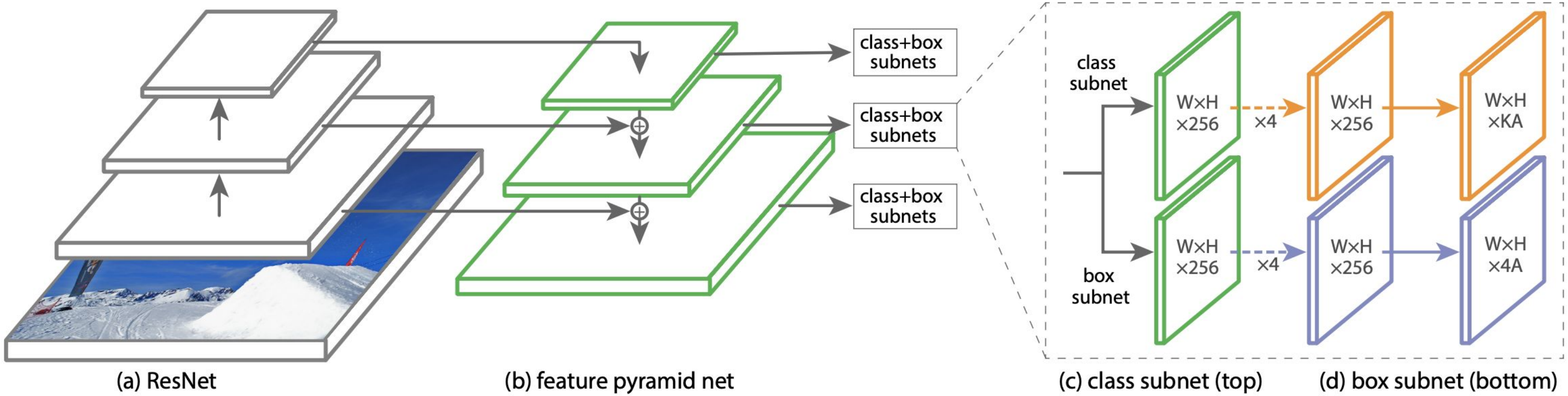


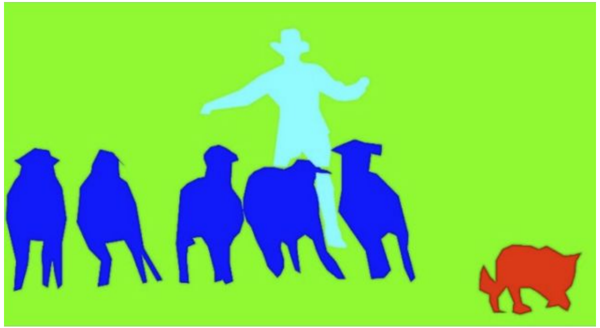
Figure from *Focal Loss for Dense Object Detection*, Lin et al, 2017

Bonus: Instance segmentation

Want to learn more?

He et al. **Mask R-CNN (2018)**

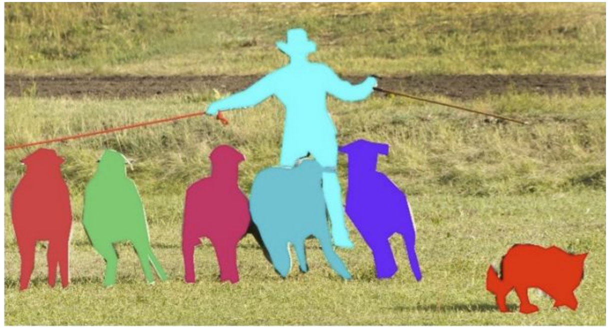
Semantic segmentation



Pixel-wise labels can be confusing for overlapping objects in the same category.

Instance segmentation

→ Object detection + segmentation



Figures from *Microsoft COCO: Common Objects in Context, Lin et al, 2014*

Metrics and benchmarks

Evaluation metrics

Want to learn more?

Berman et al. **The Lovasz-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks (2018)**

Classification

- **Accuracy**: percentage of correct predictions

Top-1: top prediction is the correct class

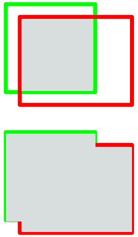
Top-5: correct class is in top-5 predictions

Object detection and segmentation

- intersection-over-union (**IoU**)

non-differentiable: used only for evaluation

$$\mathcal{J}(\mathbf{P},\mathbf{T}) = \frac{\mathbf{P} \bigcap \mathbf{T}}{\mathbf{P} \bigcup \mathbf{T}}$$

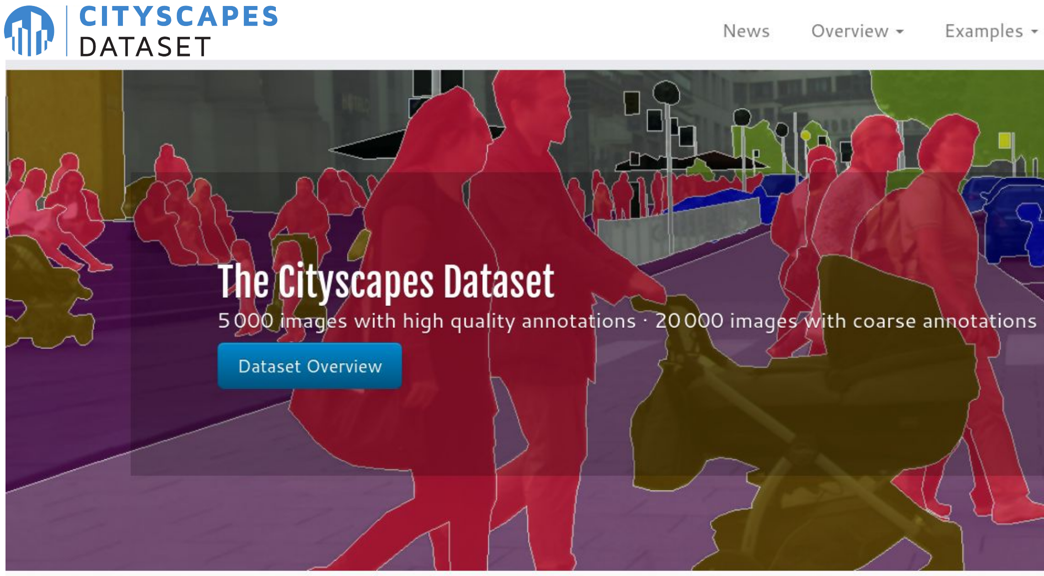


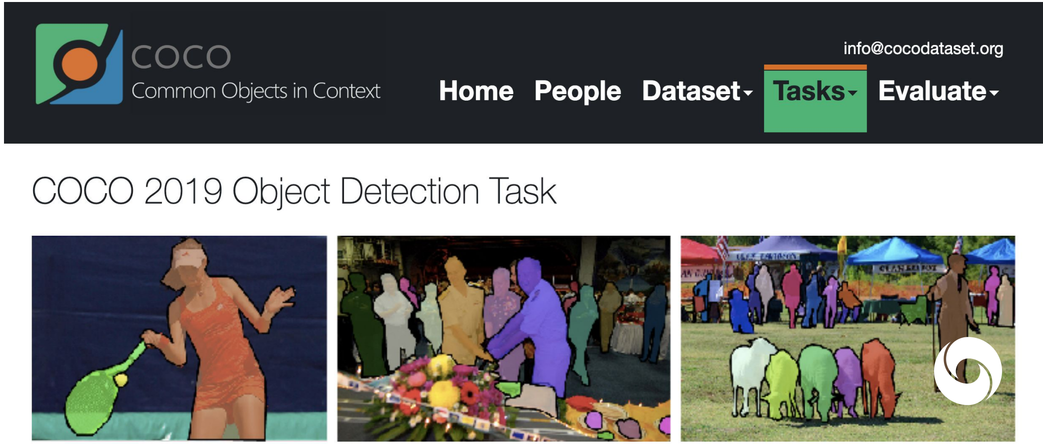
Benchmarks

Similar to Imagenet for various tasks

Public platforms for model evaluation

Maintain a leaderboard to track state-of-the-art models





Tricks of the trade

Transfer learning

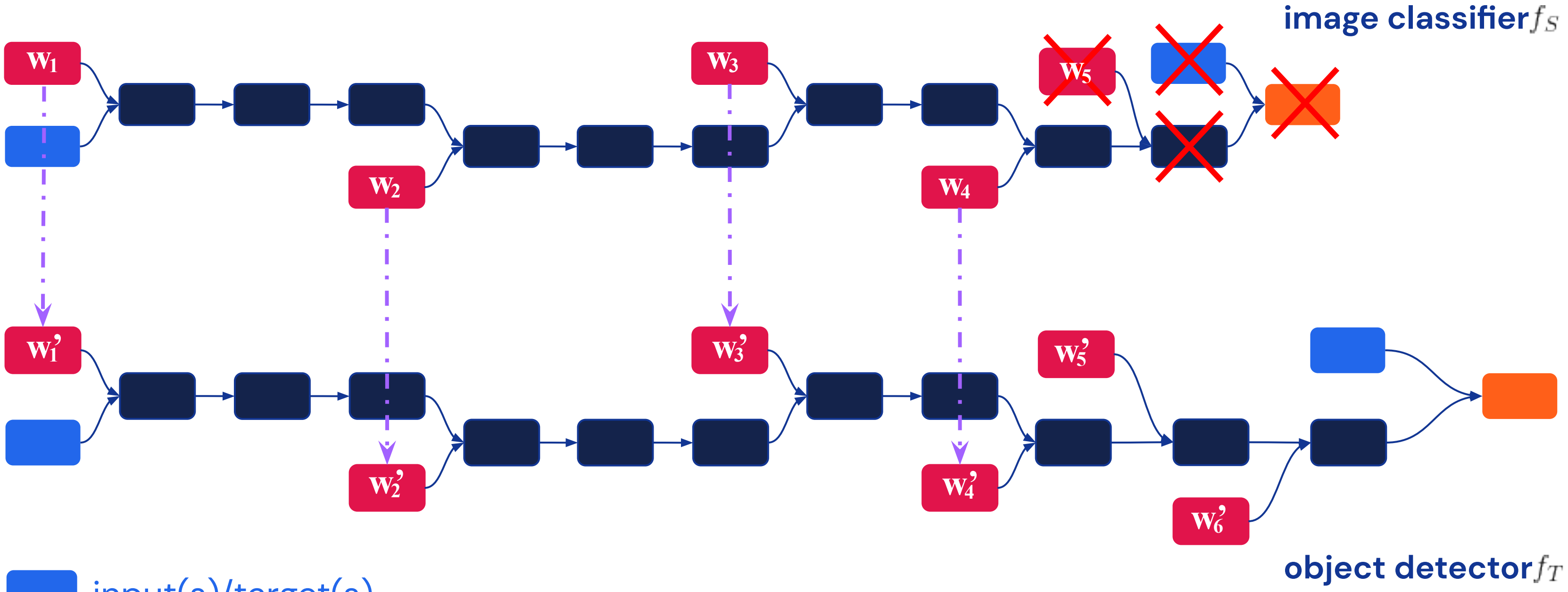
Let $\mathcal{D} = \{\mathcal{X}, P(X)\}, X = \{x_1, ..., x_n\} \in \mathcal{X}$ be a domain and $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}, f(x_i) = \hat{y}_i, y_i \in \mathcal{Y}$ a task defined on this domain.

Given a source domain and task $\begin{pmatrix}\mathcal{D}_S \\ \mathcal{T}_S\end{pmatrix}$ and a target domain and task $\begin{pmatrix}\mathcal{D}_T \\ \mathcal{T}_T\end{pmatrix}$, reuse knowledge learnt by f_S in f_T

$$\begin{pmatrix}\mathcal{D}_S \\ \mathcal{T}_S\end{pmatrix} \rightrightarrows \begin{pmatrix}\mathcal{D}_T \\ \mathcal{T}_T\end{pmatrix}$$

Intuition: features are shared across tasks and datasets. **Reuse knowledge**.

Transfer learning across different tasks



input(s)/target(s)
computation
parameters
loss

$w_{1\text{-}4}'$ fine-tuned

$w_{5\text{-}6}'$ trained from scratch

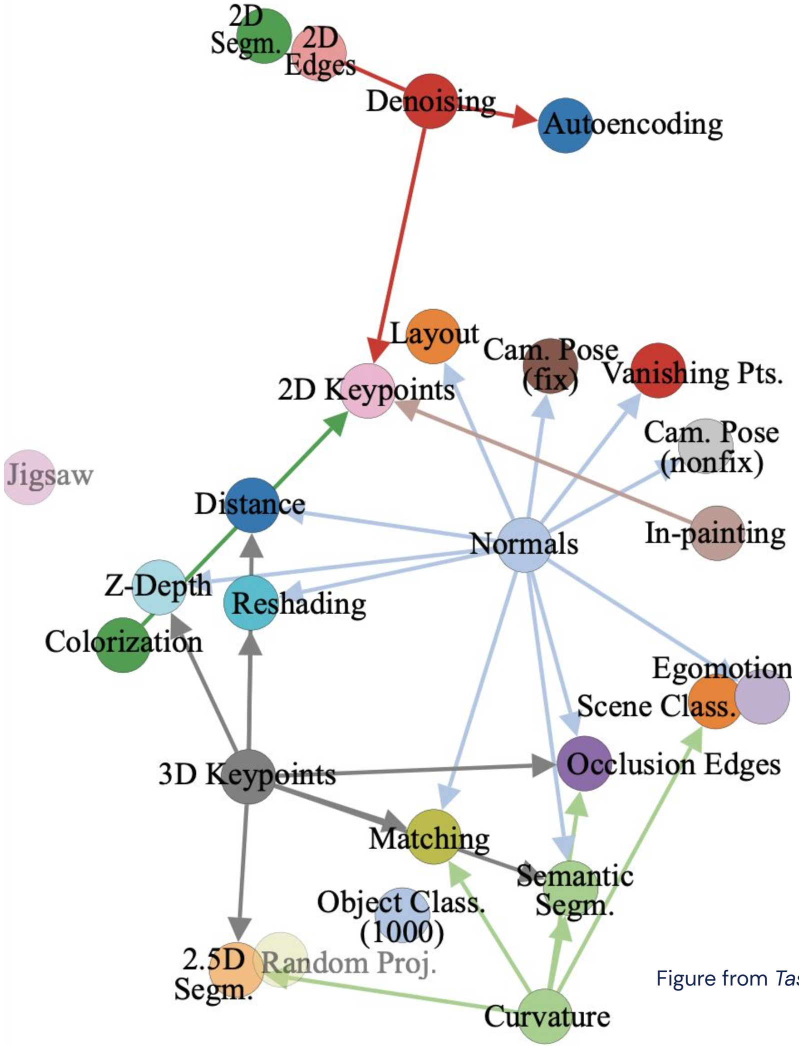


Figure from *Taskonomy: Disentangling Task Transfer Learning*, Zamir et al, 2018

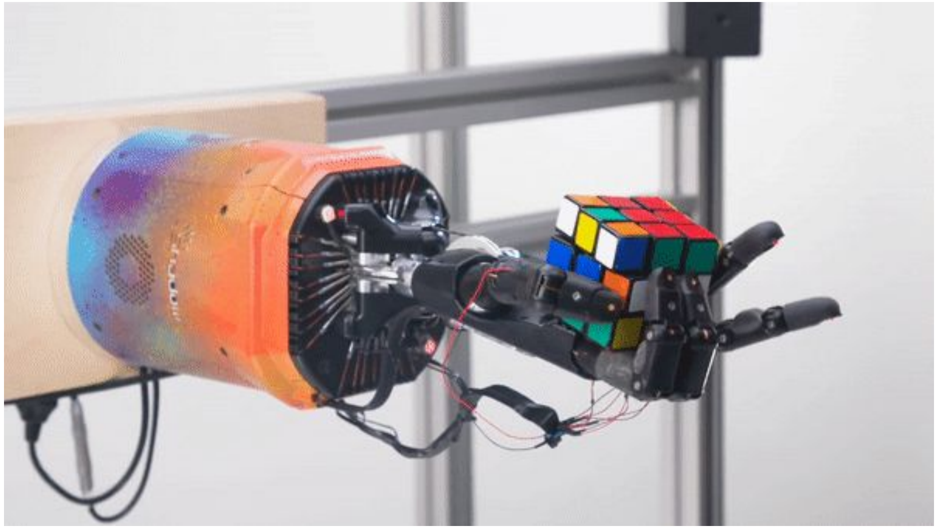
Want to learn more?

Zamir et al. **Taskonomy: Disentangling Task Transfer Learning (2018)**

Transfer learning across different domains

Sim2Real

- Train in simulation using RL – $\mathcal{D}_S$
- Use Automatic Domain Randomization: data augmentation + hard negative mining
- Test in real world – $\mathcal{D}_T$



Solving Rubik's Cube with a Robot Hand, OpenAI, 2019

Beyond supervised image classification

01 Supervised image ~~classification~~

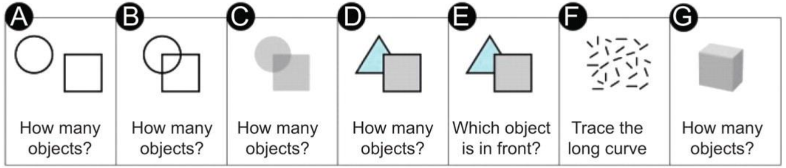
02 Supervised ~~image~~ classification

03 ~~Supervised~~ image classification

04 Open questions

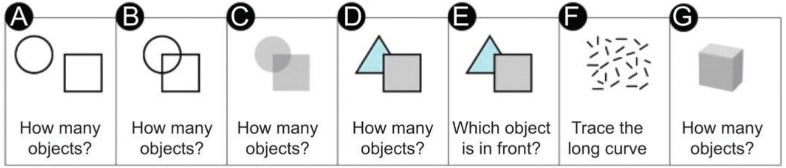
2 Beyond single image input

Experiment



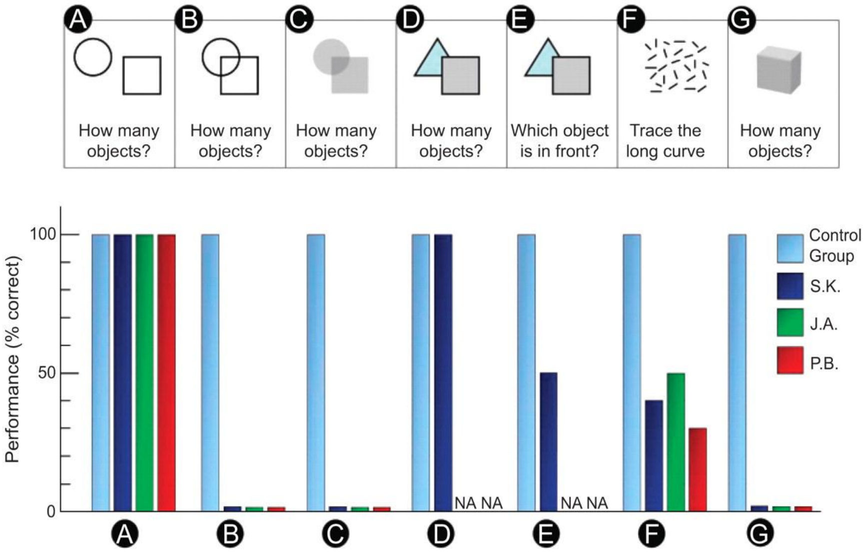
Visual Parsing After Recovery From Blindness, Ostrovsky et al, 2009

Experiment



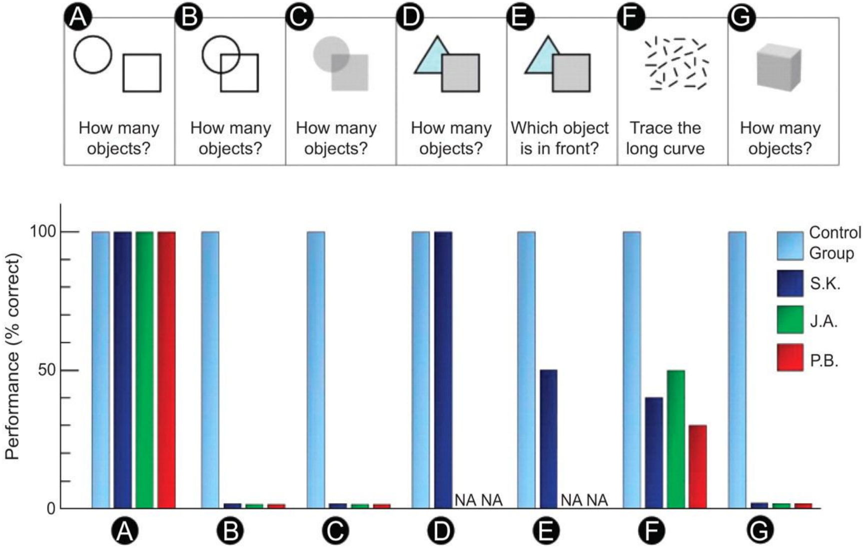
Visual Parsing After Recovery From Blindness, Ostrovsky et al, 2009

Experiment



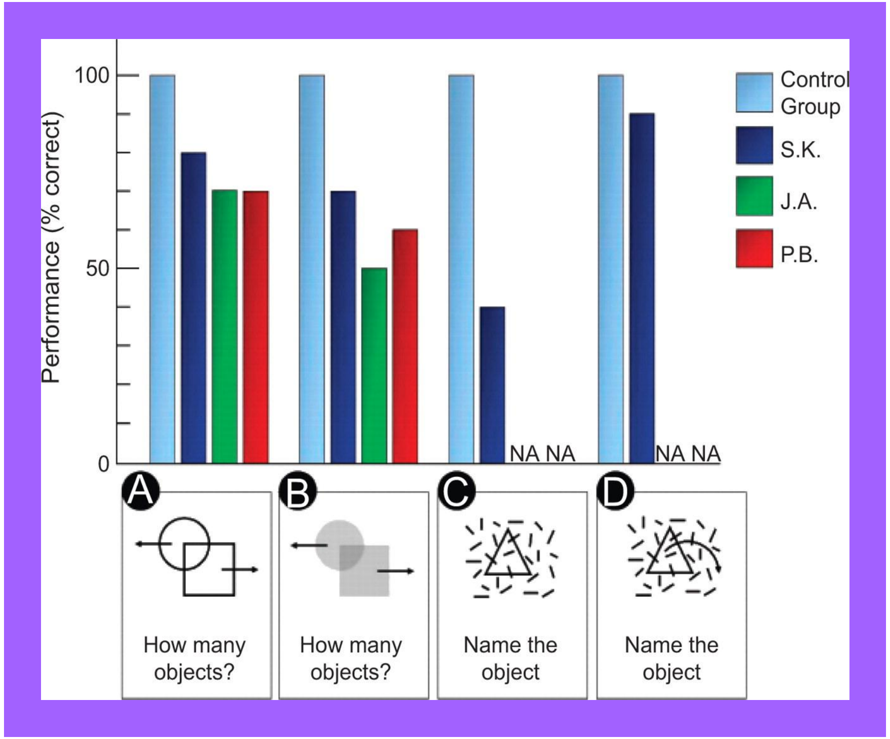
Visual Parsing After Recovery From Blindness, Ostrovsky et al, 2009

Experiment



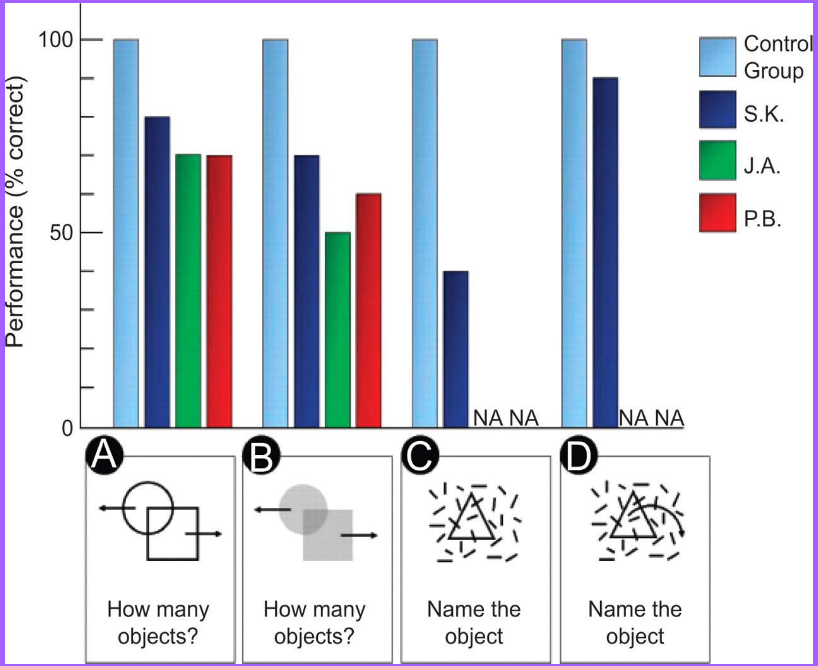
Visual Parsing After Recovery From Blindness, Ostrovsky et al, 2009

Experiment



Visual Parsing After Recovery From Blindness, Ostrovsky et al, 2009

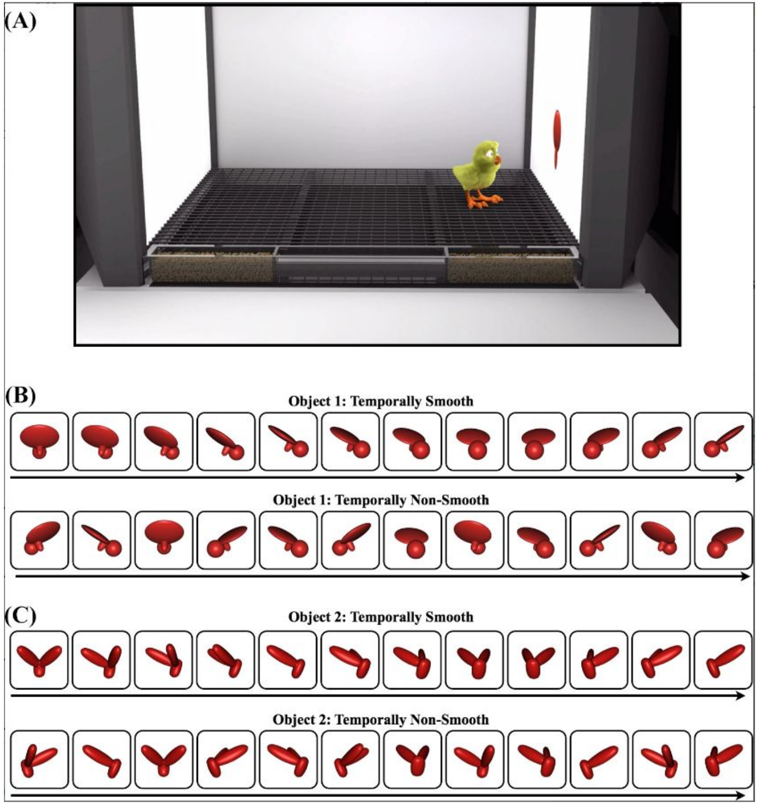
Experiment



Motion helps object recognition when learning to see.

Visual Parsing After Recovery From Blindness, Ostrovsky et al, 2009

Experiment

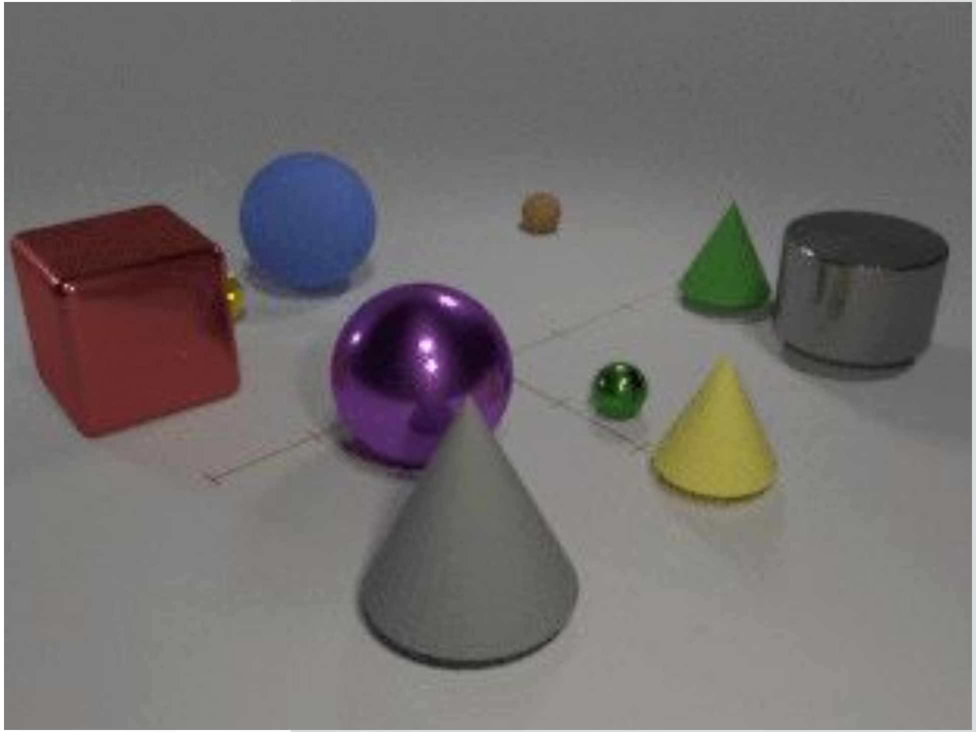


Motion helps object recognition when learning to see.

The Development of Invariant Object Recognition Requires Visual Experience With Temporally Smooth Objects, Wood and Wood, 2018

Videos

- Motion – cues for object recognition during learning
- Natural data augmentation: translation, scale, 3D rotation, camera motion, light changes



CATER: A diagnostic dataset for Compositional Actions and Temporal Reasoning, Girdhar and Ramanan, 2019

Beyond single image input

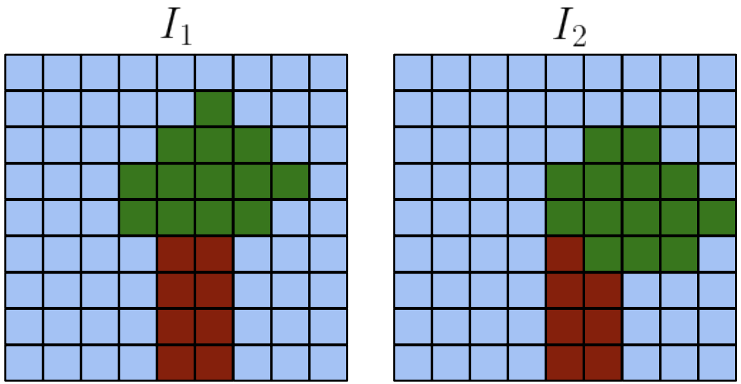
Inputs	Task definitions	Models	Challenges
Pairs of images	Optical flow estimation	Image-based models	Obtaining labels
Videos	Action recognition	3D convnets	A note on efficiency
		Recurrent (not covered)	

Pairs of images input

Optical flow estimation

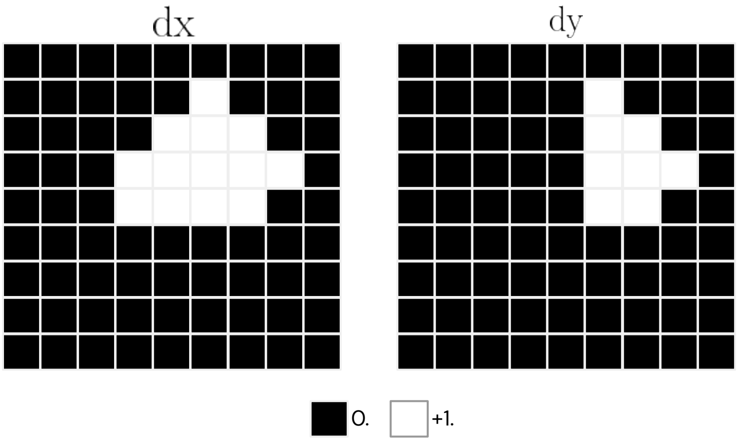
Inputs

- Pair of RGB images



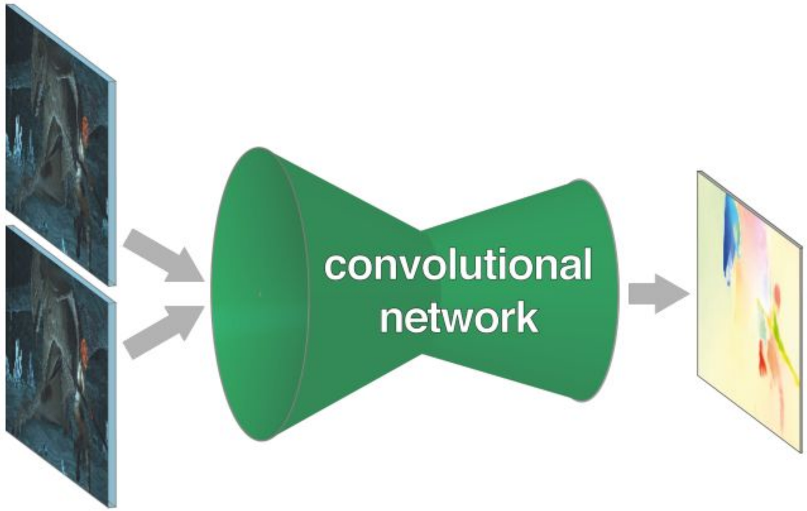
Targets

- Dense flow map (real values)
- 2D translation displacements



Case study - FlowNet

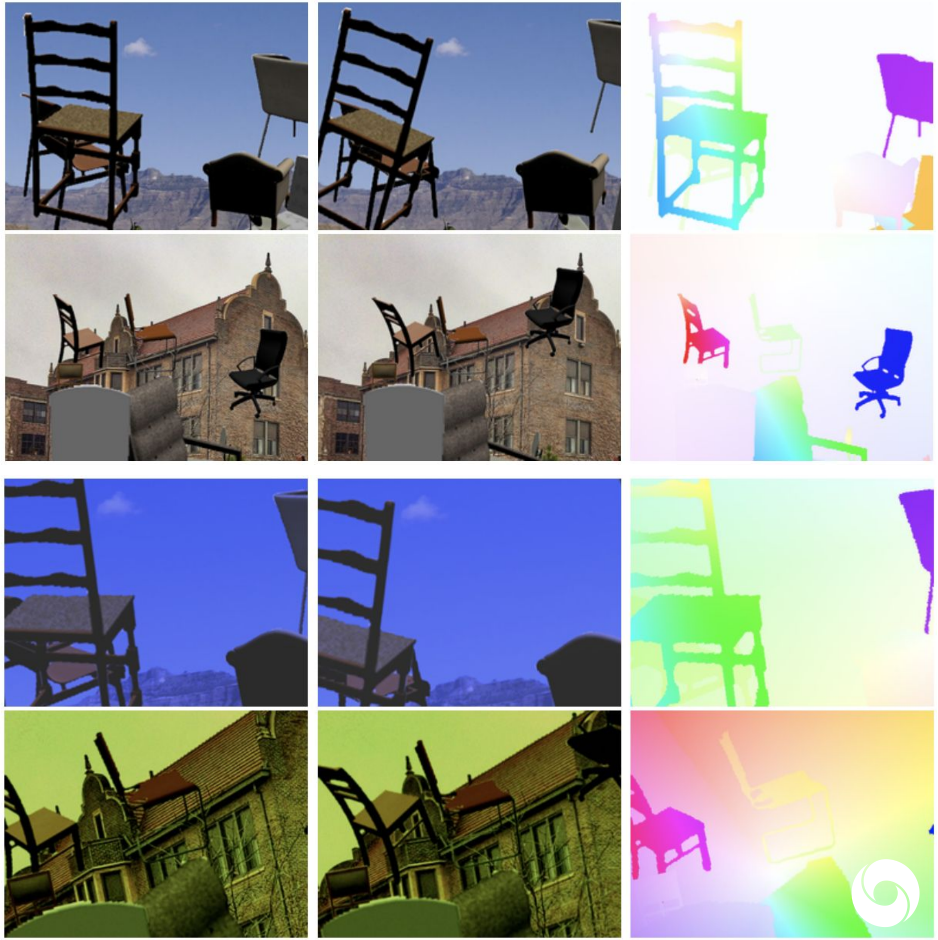
- Encoder-decoder architecture similar to U-NET
- Supervised training
- Loss: Euclidean distance
- *Flying chairs* dataset



FlowNet: Learning optical flow with convolutional networks, Fischer et al, 2015

Case study - FlowNet

- Encoder-decoder architecture similar to U-NET
- Supervised training
- Loss: Euclidean distance
- *Flying chairs* dataset
- Sim2Real transfer



FlowNet: Learning optical flow with convolutional networks, Fischer et al, 2015

Video input

Video models from image models

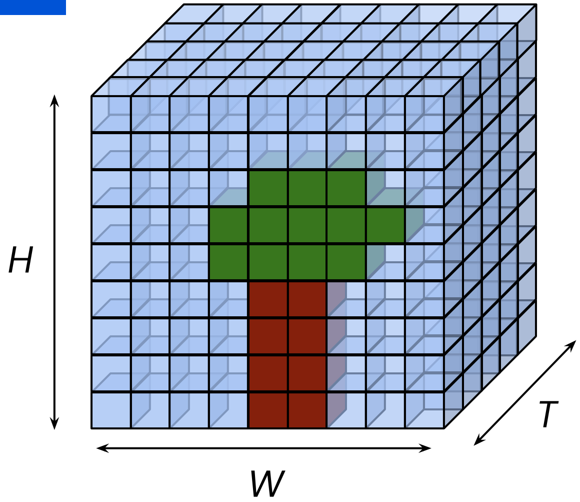
Cityscapes

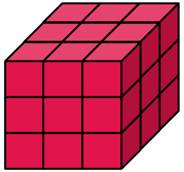
Improving Semantic Segmentation via Video Propagation and Label Relaxation, Zhu et al, 2019

Video models using 3D convolutions

Video as a volume

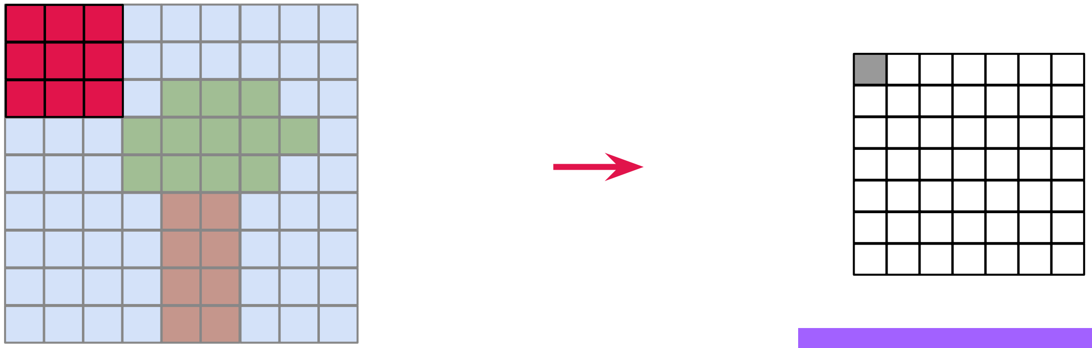
- stack frames T x H x W x 3
- apply 3D convolutions





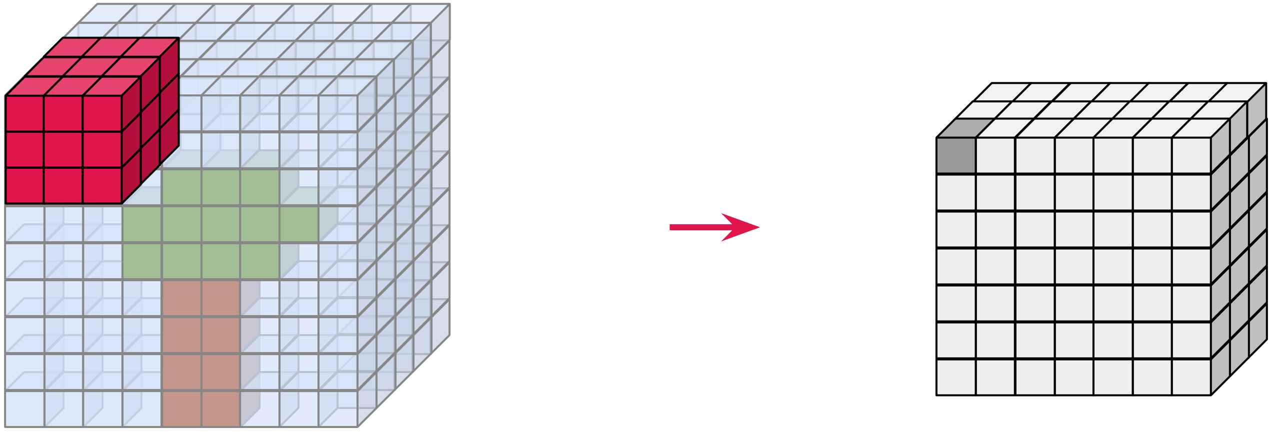
$$y = \sum_{i \in 3\times3\times3} \mathbf{w}_i \mathbf{x}_i + b$$

Recap: 2D convolution operation



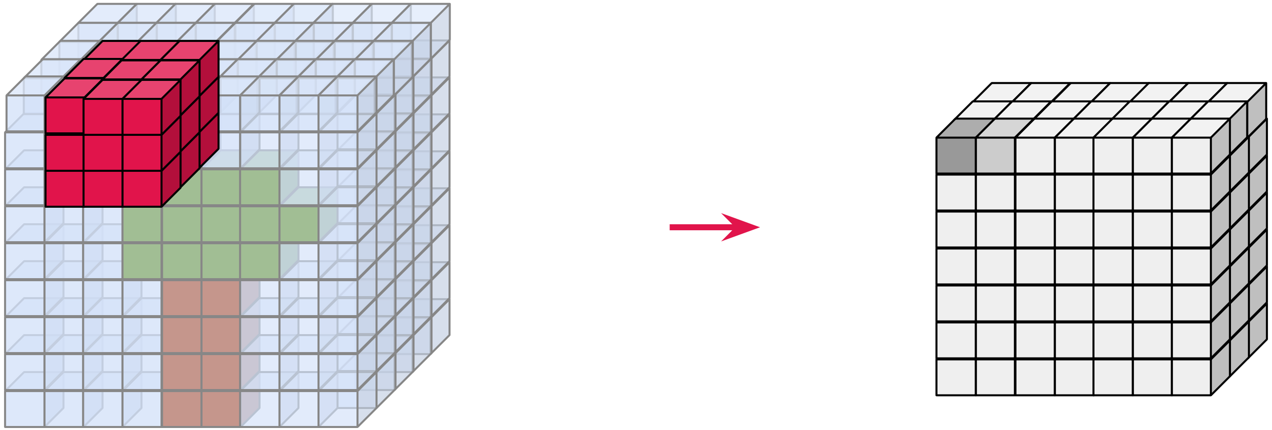
The **kernel** slides across spatial dimensions.

3D convolution operation



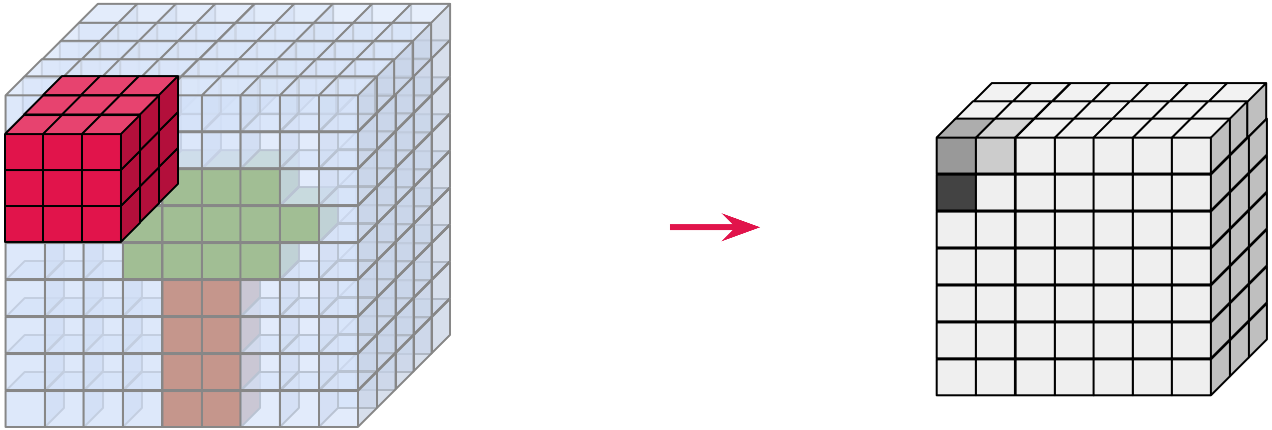
The kernel slides across **space and time** to generate spatio-temporal feature maps.

3D convolution operation



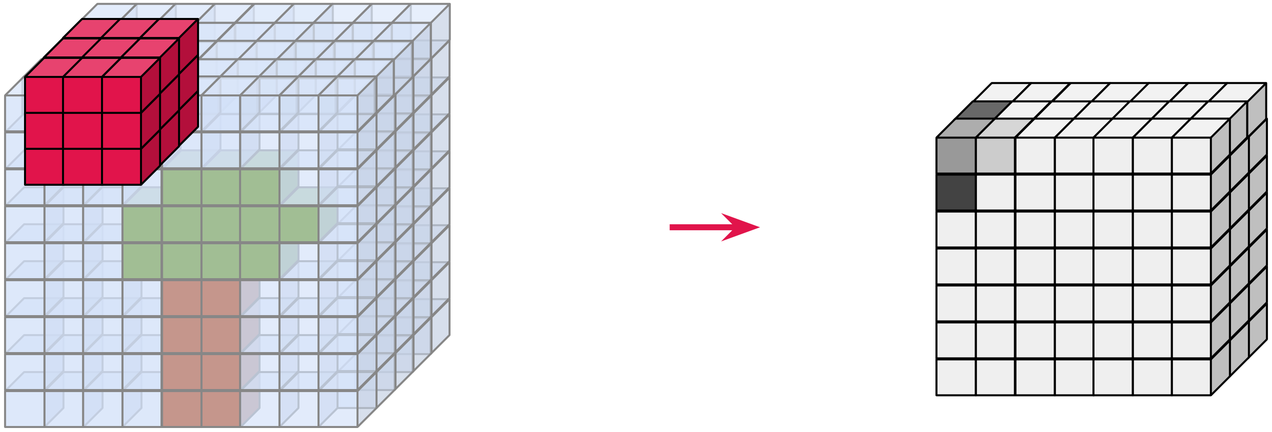
The kernel slides across **space and time** to generate spatio-temporal feature maps.

3D convolution operation



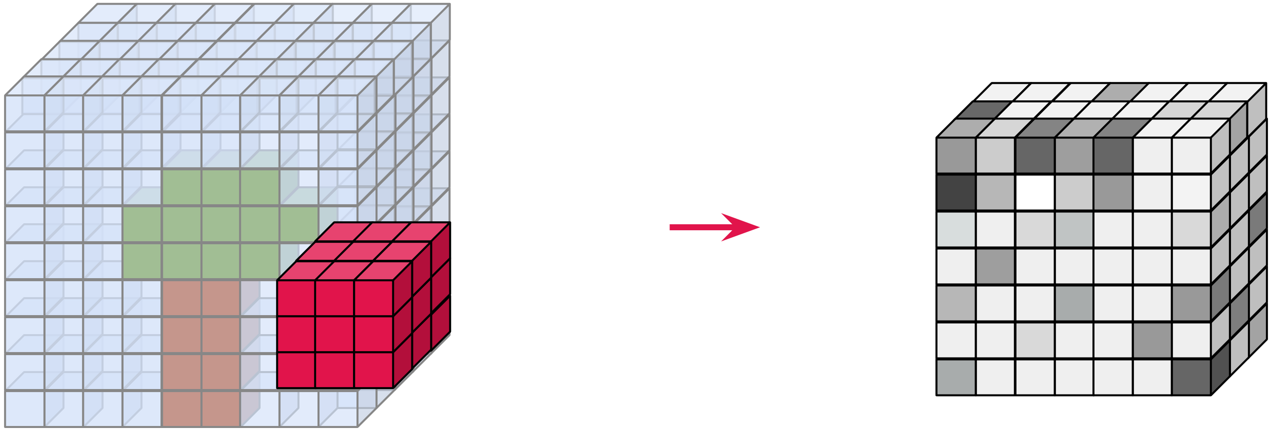
The kernel slides across **space and time** to generate spatio-temporal feature maps.

3D convolution operation



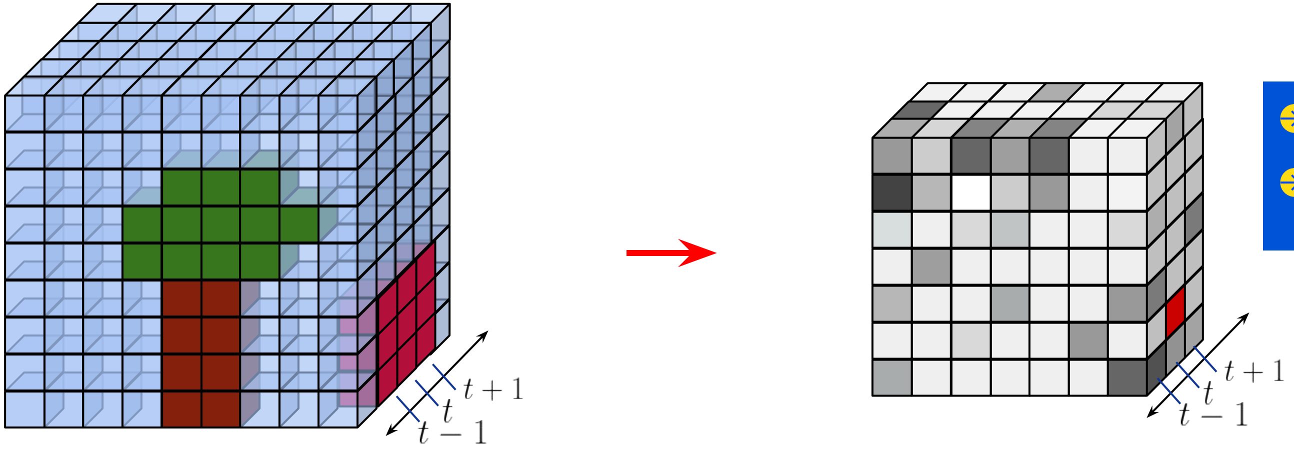
The kernel slides across **space and time** to generate spatio-temporal feature maps.

3D convolution operation



The kernel slides across **space and time** to generate spatio-temporal feature maps.

Properties of 3D convolutions



- 3D convolutions are non-causal
- masked 3D convolutions are causal

Strided, dilated, padded, [...] convolutions apply in 3D as well.

Action recognition

Inputs

- RGB video $T \times H \times W \times 3$
- (optional) flow map $T \times H \times W \times 2$

Targets

- action label one_hot $1 \times N_{classes}$

  e.g. *cricket shot*

Kinetics600 dataset

- 600k training videos, 600 classes
- Curated Youtube videos
- Each video: 250 frames (~ 10 sec.)
- Current accuracy: **81.8 % top-1**

Video from Kinetics dataset, Carreira et al, 2017

Case study: SlowFast

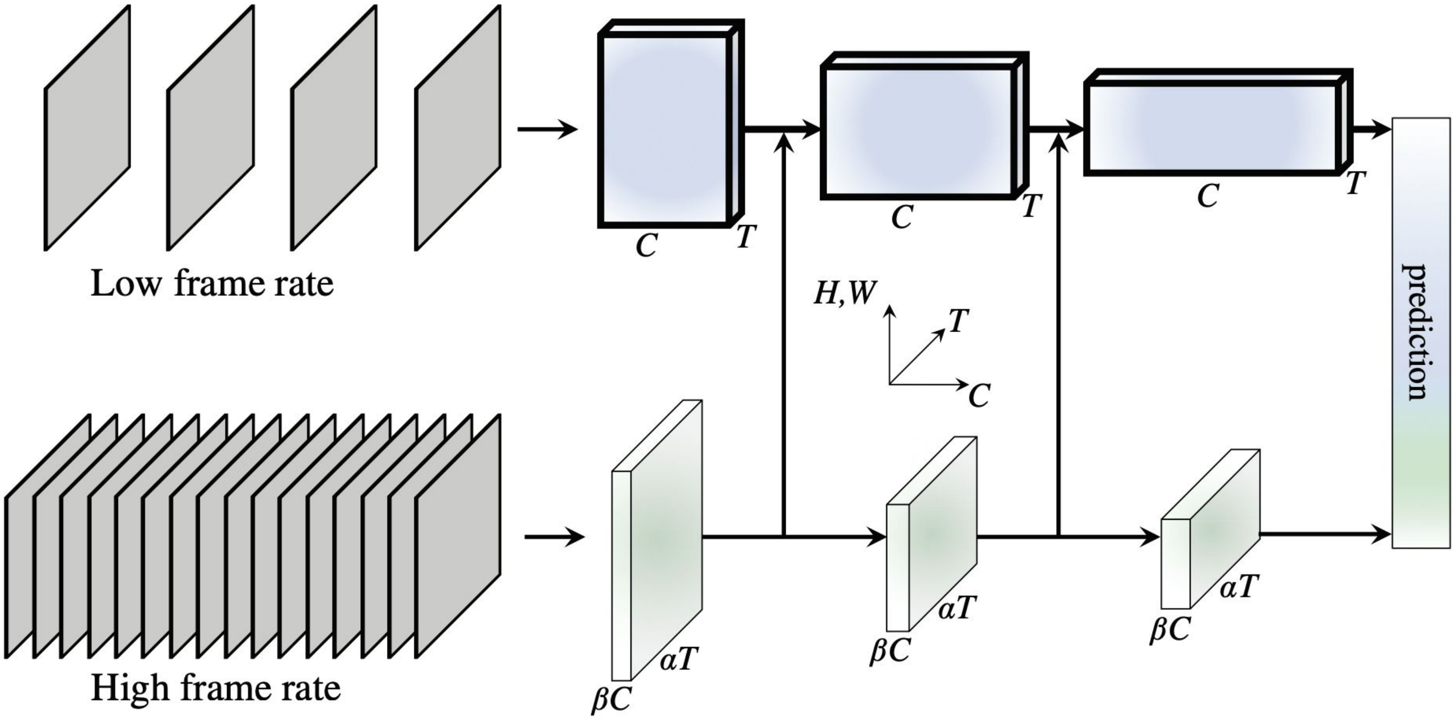


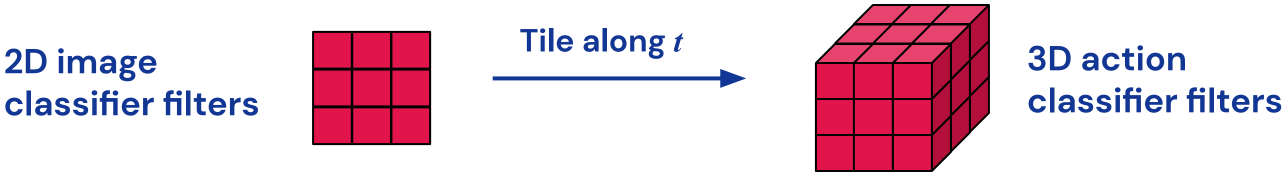
Figure from *SlowFast Networks for Video Recognition*, Feichtenhofer et al, 2019

Transfer learning returns

Want to learn more?

Carreira and Zisserman. **Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset (2017)**

Inflating 2D kernels into 3D



Intuition: a tiled image is a video of a static scene, filmed with a fixed camera.

Challenges

Challenges in video processing

- Difficult to obtain labels
- Large memory requirements
- High latency
- High energy consumption

Improve efficiency of video models

- Inspiration from biological systems
- Maximise parallelism to increase throughput and reduce latency [1, 2]
- Exploit redundancies in the visual data to obtain frugal models [3]

[1] *Massively parallel video networks*, Carreira, Patraucean et al, 2018
[2] *Sideways: depth-parallel training of video models*, Malinowski, Swirszcz, Carreira, Patraucean, 2020
[3] *Blink and you won't miss it: video processing without temporal redundancies*, Patraucean et al, 2020

Beyond supervised image classification

01 Supervised image ~~classification~~

02 Supervised ~~image~~ classification

03 ~~Supervised~~ image classification

04 Open questions

3 Beyond strong supervision

Labelling is tedious - Research topic in itself

PolygonRNN++

Interactive Object Annotation with Polygons

NOTE: If inference is slow due to heavy traffic (benchmark is 0.3 seconds per interaction), please consider trying our demo locally using our available code

For sponsorship/donation to help develop this web tool, please contact polyrnn@cs.toronto.edu

Tutorial

New Object (n)

Finish Object (f)

Discard Object (d/del)

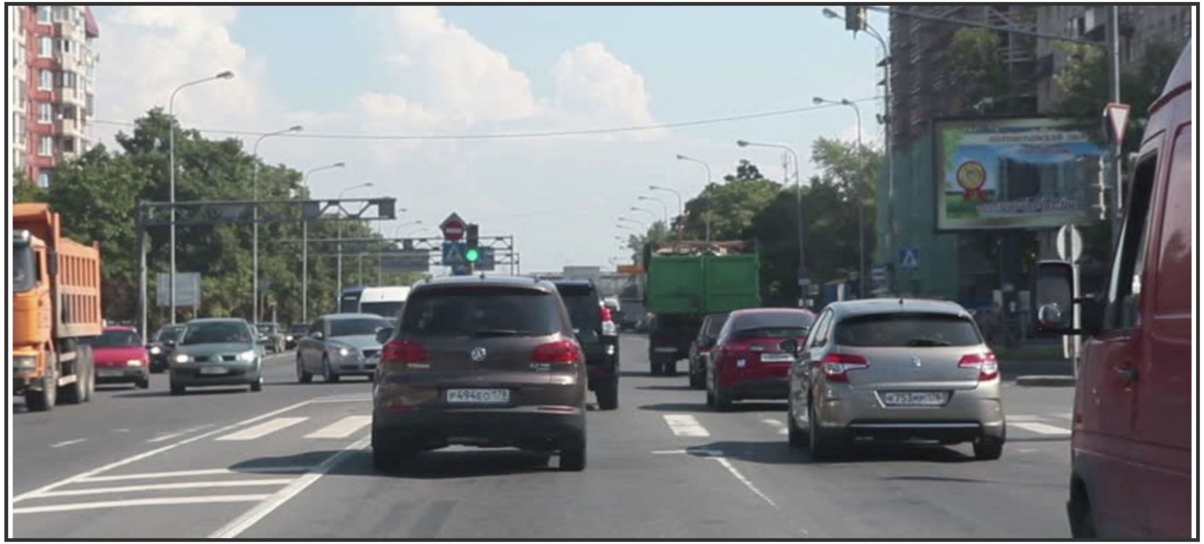
AI Smooth Poly (s)

AI Assistance (a)

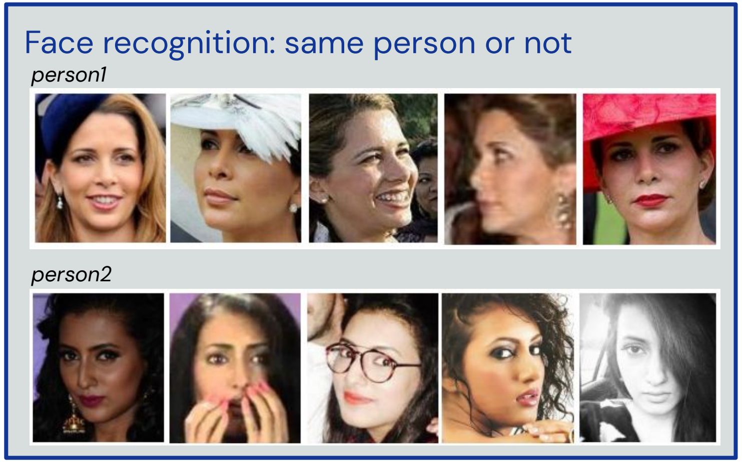
Use fine polygon (g)

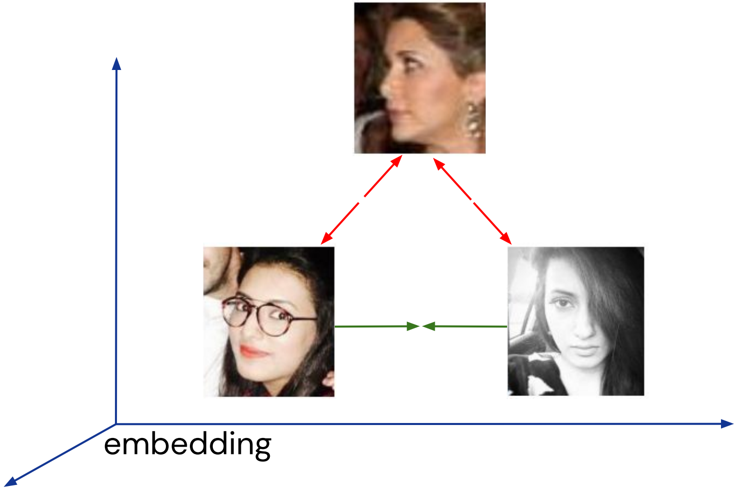
Backend:

Cityscapes



Self-supervision - Metric learning





Standard losses (e.g. cross-entropy, mean square error)

- learn mapping between input(s) and output distribution / value(s)

Metric learning

- learn to predict distances between inputs given some similarity measure (e.g. same person or not)

Images from *VGGFace2: A dataset for recognising faces across pose and age*, Cao et al, 2018

Self-supervision - Metric learning

Metric learning

- Contrastive loss
- Triplet loss
- State-of-the-art on representation learning

Applications

- (Multimodal) self-supervised representations, e.g. image+sound [1]
- Information retrieval [2]
- Low-shot face recognition [3]

[1] *Look, listen and learn,* Arandjelovic and Zisserman, 2017
[2] *Learning to Learn from Web Data through Deep Semantic Embeddings,* Gomez et al, 2018
[3] *VGGFace2: A dataset for recognising faces across pose and age,* Cao et al, 2018

Metric learning

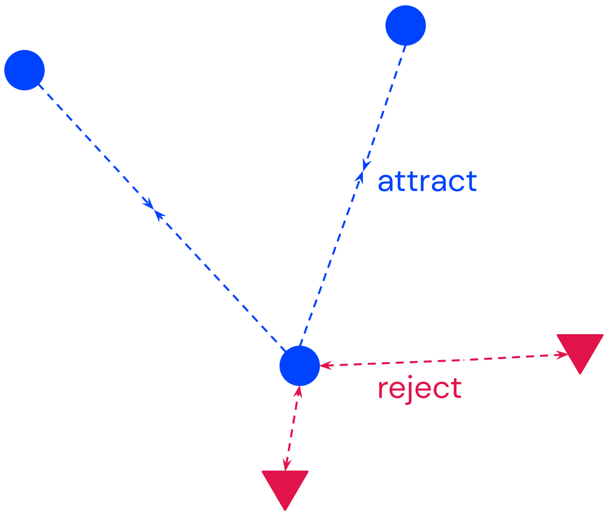
Contrastive loss (margin loss)

Dataset:

$$(r_0, r_1, y) \qquad \begin{cases} y = 1, & \text{if } (r_0, r_1) \text{ same-person} \\ y = 0, & \text{otherwise} \end{cases}$$

$$\ell(r_0, r_1, y) = y\mathrm{d}(r_0, r_1)^2 + (1-y)(\max(0, m - \mathrm{d}(r_0, r_1)))^2$$

$\mathrm{d}(\cdot,\cdot)$ – Euclidean distance



Dimensionality reduction by learning an invariant mapping, Hadsell et al, 2006

Metric learning

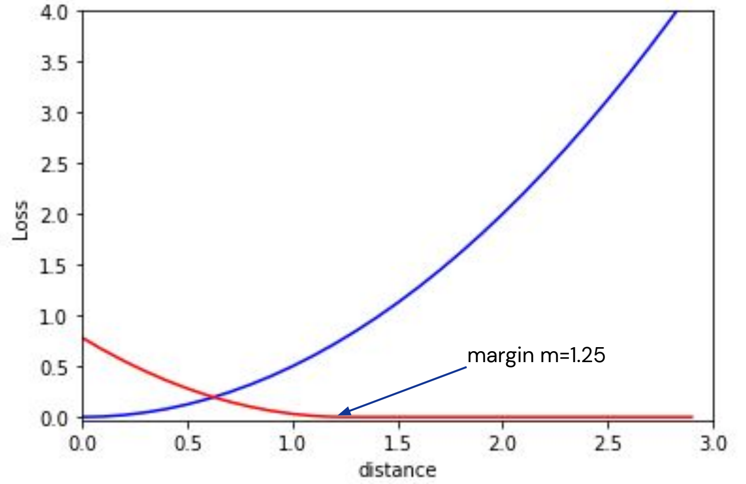
Contrastive loss (margin loss)

Dataset:

$$(r_0, r_1, y) \qquad \begin{cases} y = 1, & \text{if } (r_0, r_1) \text{ same-person} \\ y = 0, & \text{otherwise} \end{cases}$$

$$\ell(r_0, r_1, y) = y\mathrm{d}(r_0, r_1)^2 + (1 - y)(\max(0, m - \mathrm{d}(r_0, r_1)))^2$$

$\mathrm{d}(\cdot,\cdot)$ – Euclidean distance



Dimensionality reduction by learning an invariant mapping, Hadsell et al, 2006

Metric learning

Contrastive loss (margin loss)

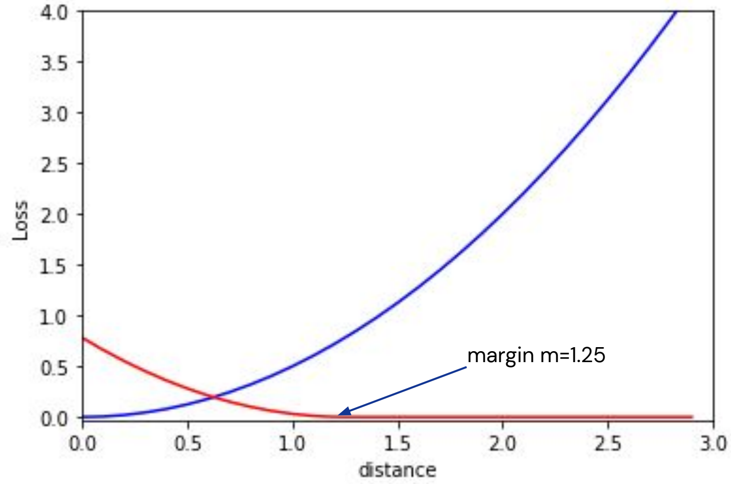
Dataset:

$$(r_0, r_1, y) \qquad \begin{cases} y = 1, & \text{if } (r_0, r_1) \text{ same-person} \\ y = 0, & \text{otherwise} \end{cases}$$

$$\ell(r_0, r_1, y) = y\mathrm{d}(r_0, r_1)^2 + (1 - y)(\max(0, m - \mathrm{d}(r_0, r_1)))^2$$

$\mathrm{d}(\cdot, \cdot)$ – Euclidean distance

Difficult to choose m



Dimensionality reduction by learning an invariant mapping, Hadsell et al, 2006

Metric learning

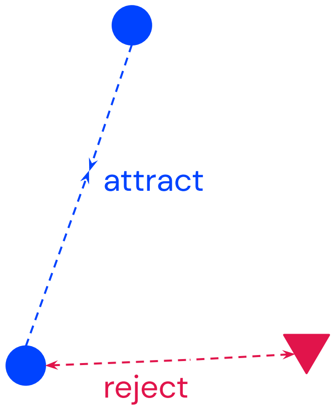
Triplet loss

Dataset:

$$(r_a, r_p, r_n) \qquad \begin{cases} (r_a, r_p) & \text{similar} \\ (r_a, r_n) & \text{dissimilar} \end{cases}$$

$$\ell(r_a, r_p, r_n) = \max(0, m + \mathrm{d}(r_a, r_p)^2 - d(r_a, r_n)^2)$$

- better than contrastive loss
- relative distances more meaningful than a fixed margin



Metric learning

Want to learn more?

Wu et al. **Sampling Matters in Deep Embedding Learning (2018)**

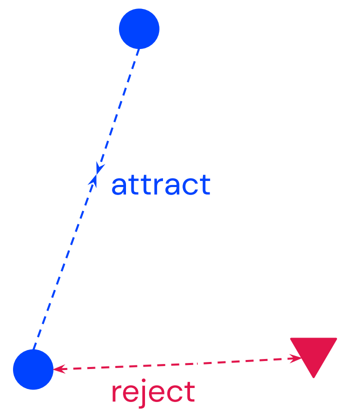
Triplet loss

Dataset:

$$(r_a, r_p, r_n) \qquad \begin{cases} (r_a, r_p) & \text{similar} \\ (r_a, r_n) & \text{dissimilar} \end{cases}$$

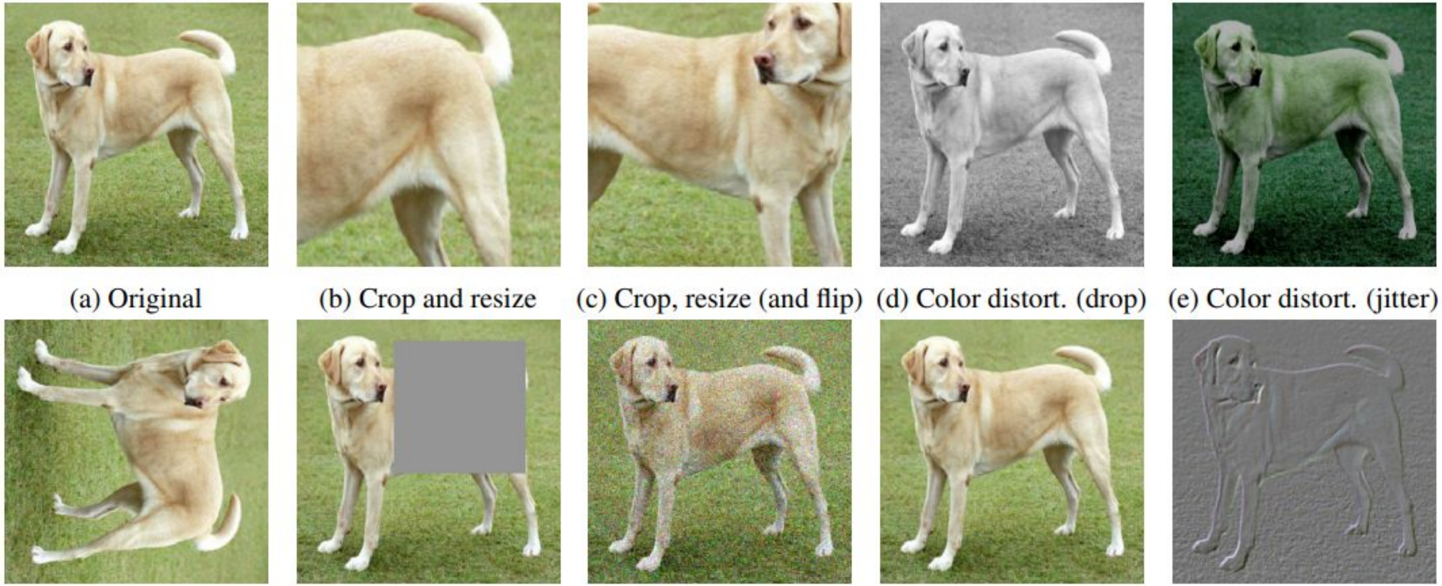
$$\ell(r_a, r_p, r_n) = \max(0, m + \mathrm{d}(r_a, r_p)^2 - d(r_a, r_n)^2)$$

- better than contrastive loss
- relative distances more meaningful than a fixed margin
- **hard negative mining** to select informative triplets



New state-of-the-art in representation learning

Same data, different augmentations



(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)
(f) Rotate {90°, 180°, 270°} (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Figure from *A Simple Framework for Contrastive Learning of Visual Representations*, Chen et al, 2020

New state-of-the-art in representation learning

- Composition of data augmentations
- Learnable non-linear transformation
- Larger mini-batches and longer training

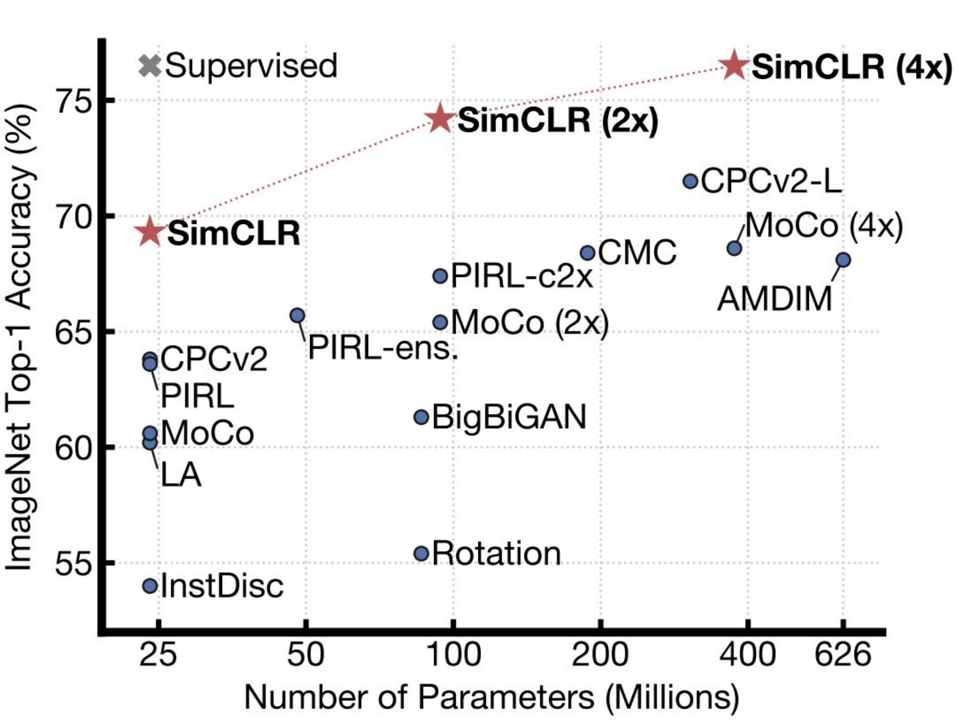


Figure from *A Simple Framework for Contrastive Learning of Visual Representations*, Chen et al, 2020

Beyond supervised image classification

01 Supervised image ~~classification~~

02 Supervised ~~image~~ classification

03 ~~Supervised~~ image classification

04 Open questions

4 Open questions

Open questions

- **Is vision solved? What does it mean to solve vision?**

  human level scene understanding – what benchmarks?

- **How to scale systems up?**

  model parallelism, better hardware, less supervision – more common sense

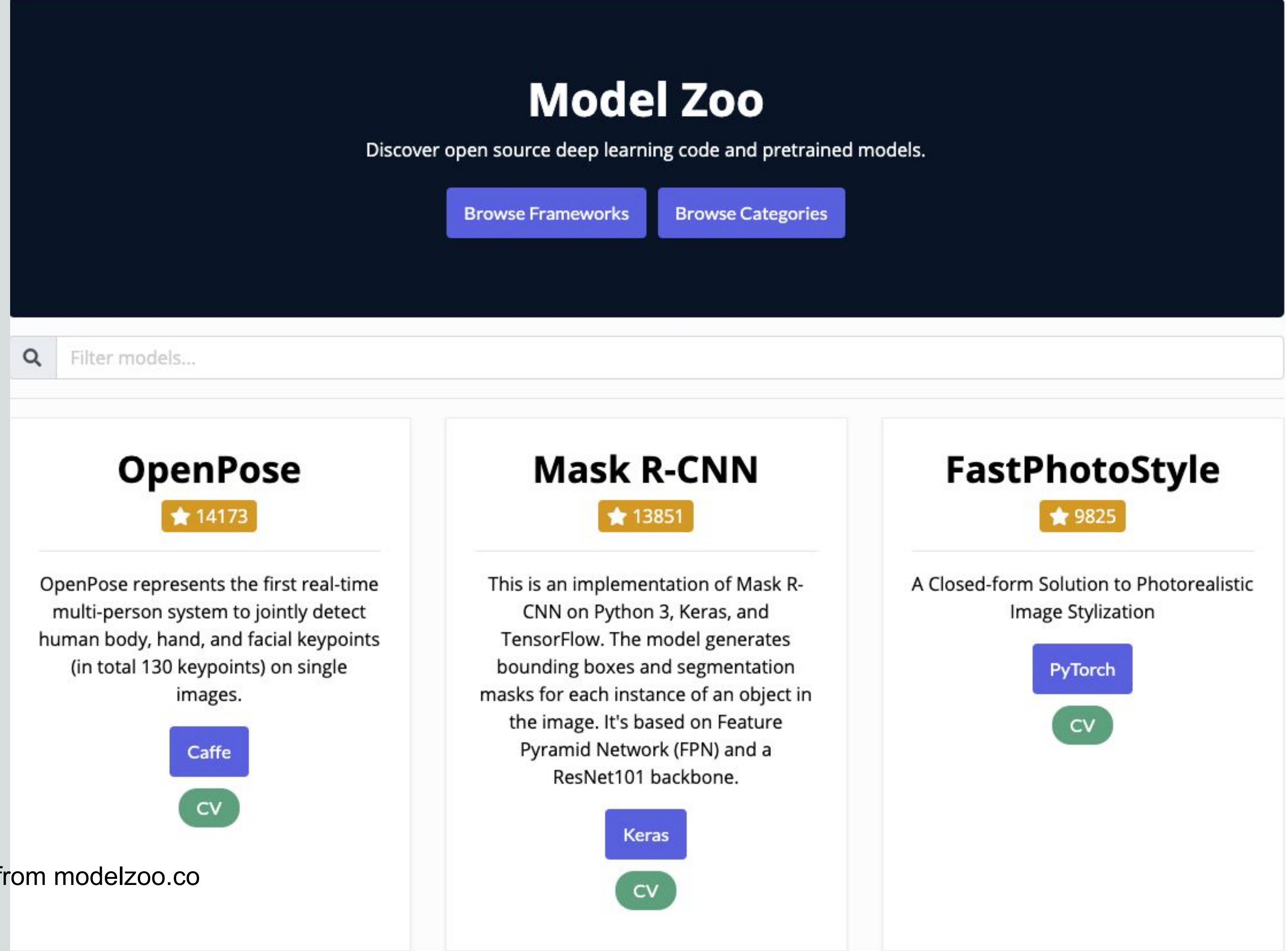
- **What are good visual representations for action?**

  Unsupervised Learning of Object Keypoints for Perception and Control, Kulkarni, Gupta et al, 2019

Learning to see from static images might make things harder than they should be.

Rethink vision models design and training from the perspective of moving pictures and with the end-goal in mind: intelligent agents that interact with the real world in real time.

Useful resources



Screenshot from modelzoo.co

Allen Institute for AI

Computer Vision Explorer

Recognition

Classification

Detection

Segmentation

Vision and Language

Human Centric Vision

Pose Estimation

Scene Geometry

Depth

Surface Normals

About

Classification

Image classification is the task of assigning an input image, a single label drawn from a fixed set of categories. Image classification models are trained and evaluated on large classification datasets such as ImageNet that has 1000 image categories.

TRY IT FOR YOURSELF

1. Choose an Image

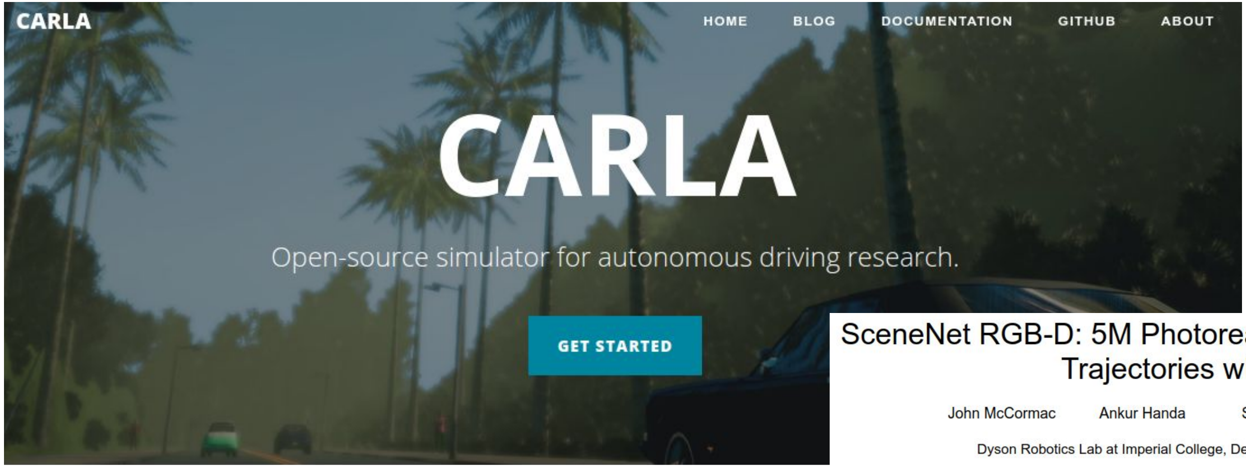
Example Images

Image:

Upload an Image

2. Run a model

Synthetic datasets for Computer Vision



SceneNet RGB-D: 5M Photorealistic Images of Synthetic Indoor Trajectories with Ground Truth

John McCormac   Ankur Handa   Stefan Leutenegger   Andrew J. Davison

Dyson Robotics Lab at Imperial College, Department of Computing, Imperial Collge London

https://www.datasetlist.com/

Transporter architecture

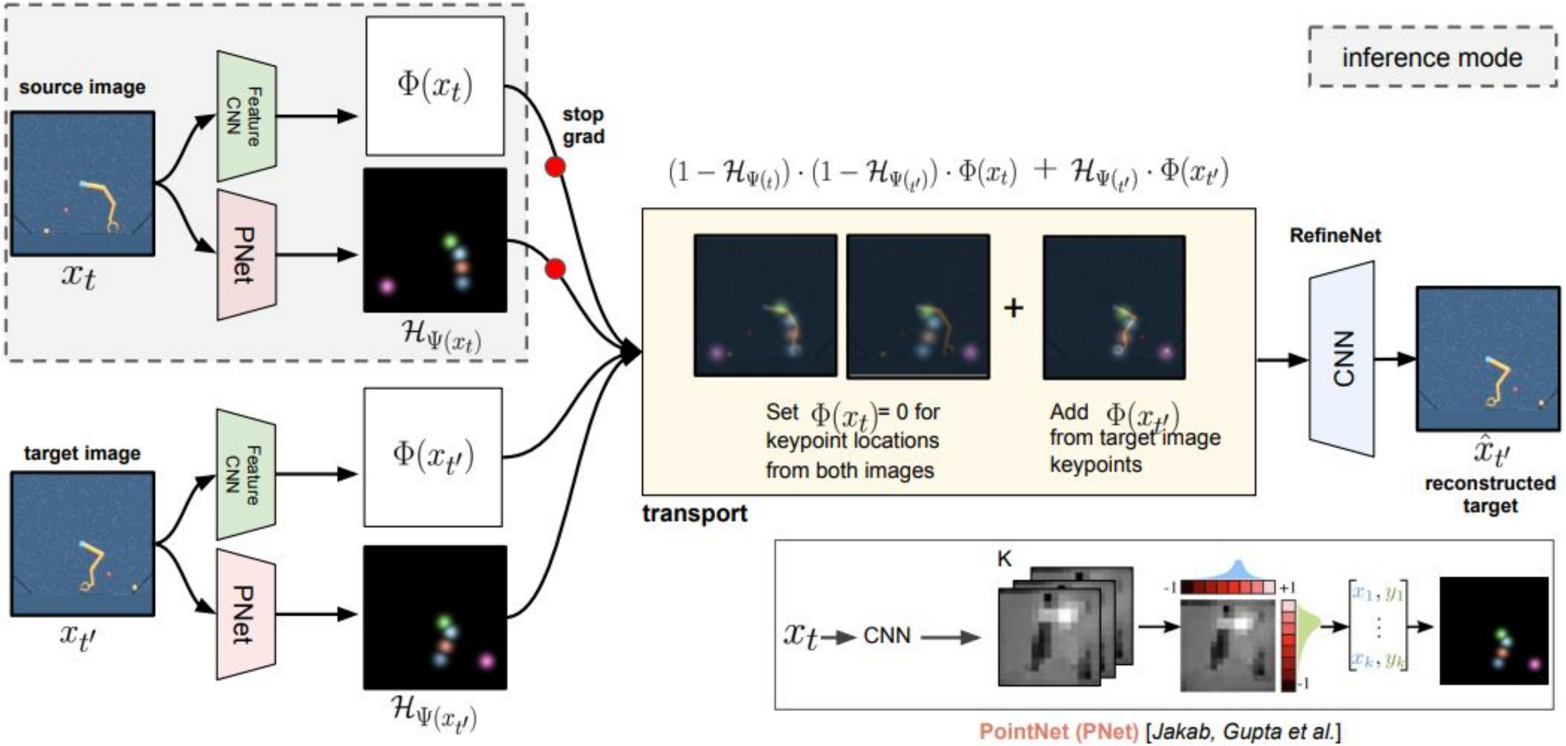


Figure from *Unsupervised Learning of Object Keypoints for Perception and Control*, Kulkarni, Gupta et al, 2019