Deep Convolutional Neural Networks II.

Jan Čech

Lecture Outline

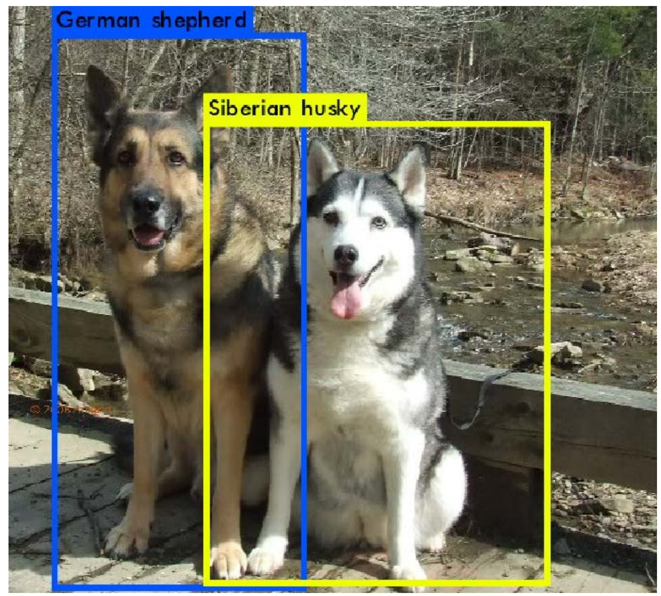
1. Deep convolutional networks for object detection
2. Some insights into the Deep Nets
3. What was not mentioned…

Deep Convolutional Networks for Object Detection

Convolutional Networks for Object Detection

- What is the object detection?

Grocery store



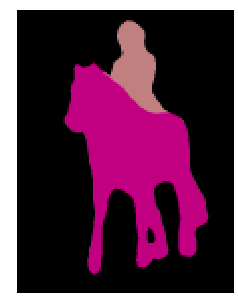


Image recognition
- What?
- holistic

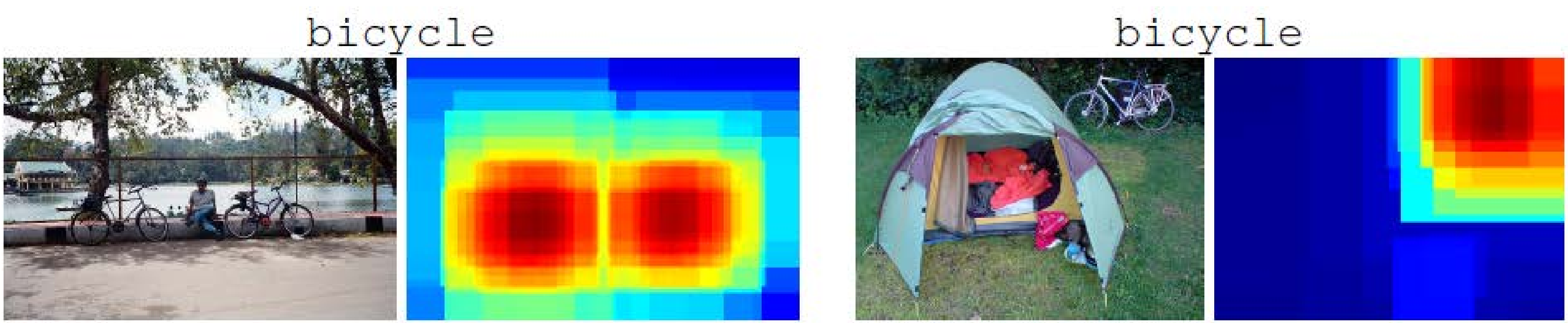
Object detection
- What + Where?
- Bounding boxes

semantic segmentation
- What + Where?
- Pixel-level accuracy

1. Scanning window + CNN

- CNN - Outstanding recognition accuracy of holistic image recognition accuracy [Krizhevsky-NIPS-2012]

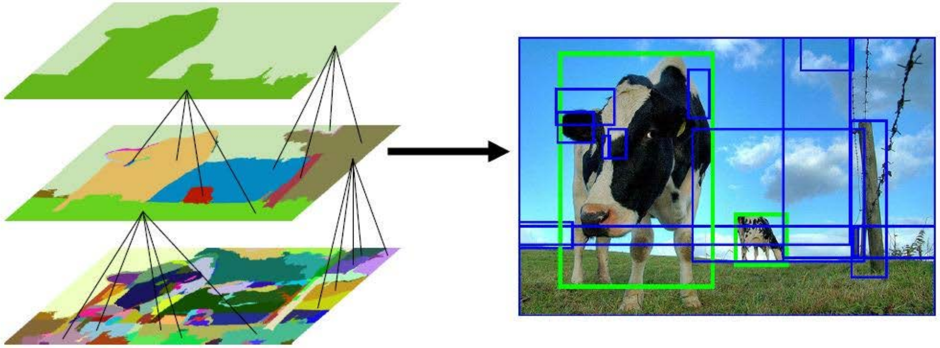
- A trivial detection extension - exhaustive scanning window
  1. Scan all possible bounding boxes
  2. Crop bounding box, warp to 224x224 (fixed-size input image)
  3. Run CNN
- Works, but
  – prohibitively slow…



Oquab et al. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks, CVPR, 2014.

2. Region proposals + CNN

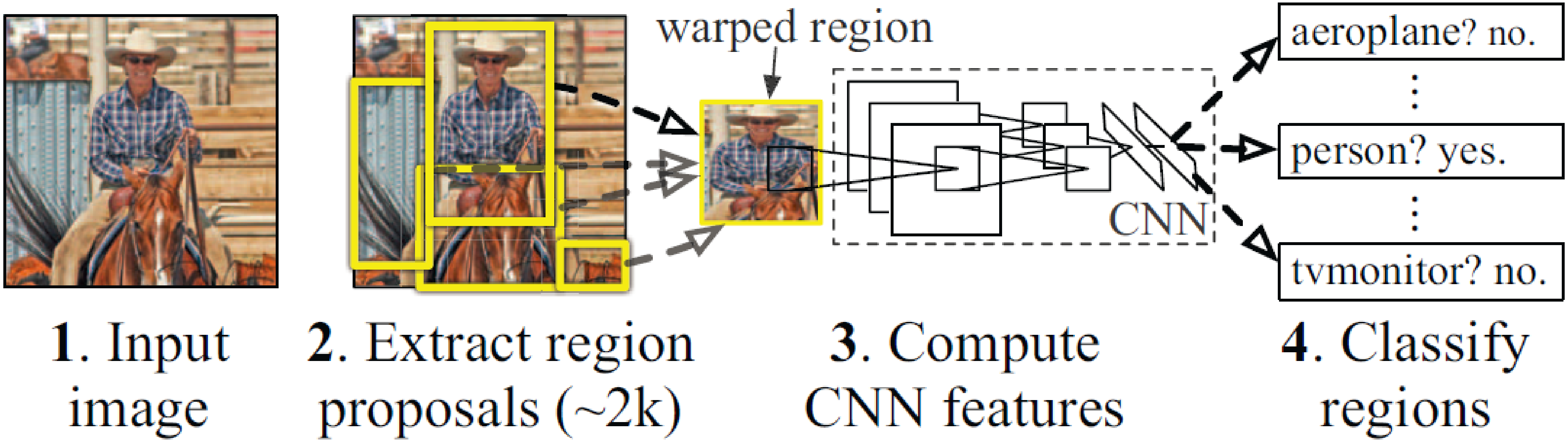
- CNN not evaluated exhaustively, but on regions where objects are likely to be present
- Region proposals (category independent):
  - Selective search [Uijlings-IJCV-2013]



  - Edgeboxes [Zitnick-ECCV-2014]

2. Region proposals + CNN

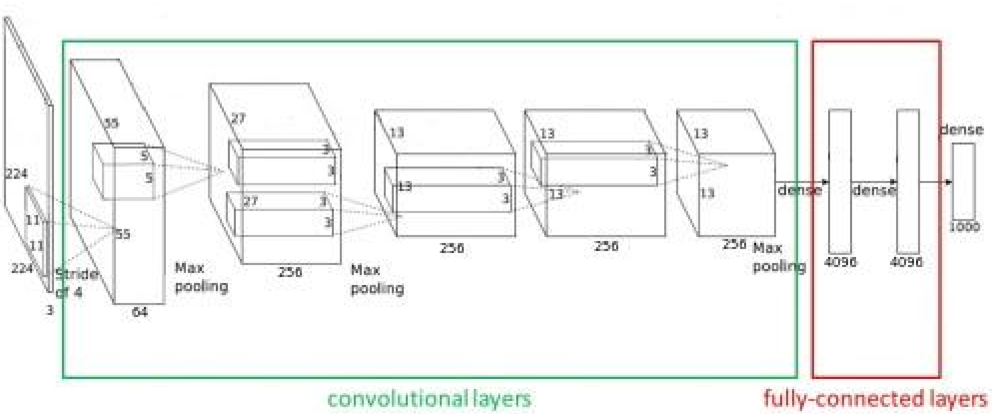
- R-CNN "Regions with CNN feature"
  - Girshick et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. CVPR 2014.

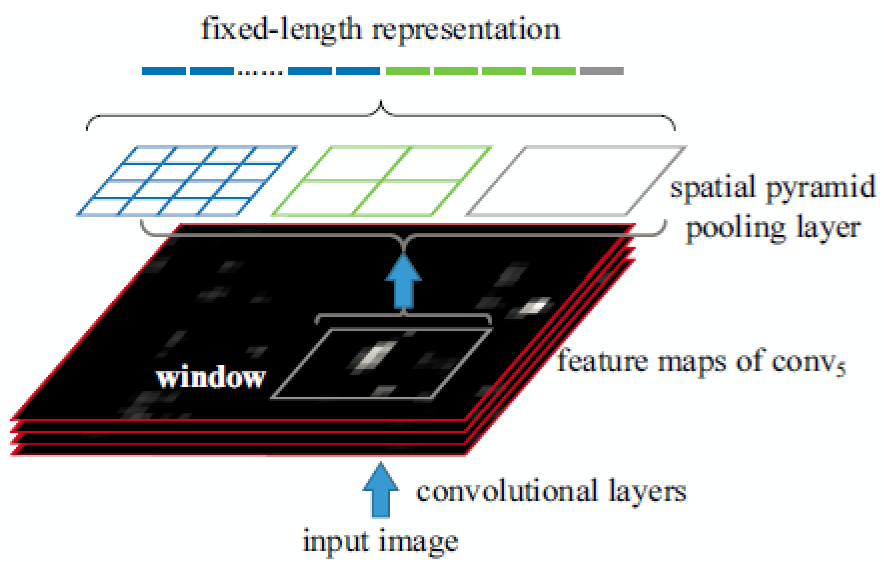


- Highly improved SotA on Pascal VOC 2012 by more than 30% (mAP)
- Still slow
  - For each region: crop + warp + run CNN (~2k)
  - 47 s/image

2. Region proposals + CNN

- Idea (1):
  - Do not run the entire CNN for each ROI, but
    - run convolutional (representation) part once for the entire image and
    - for each ROI pool the features and run fully connected (classification) part
  - He et al. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recogniton.* ECCV 2014.



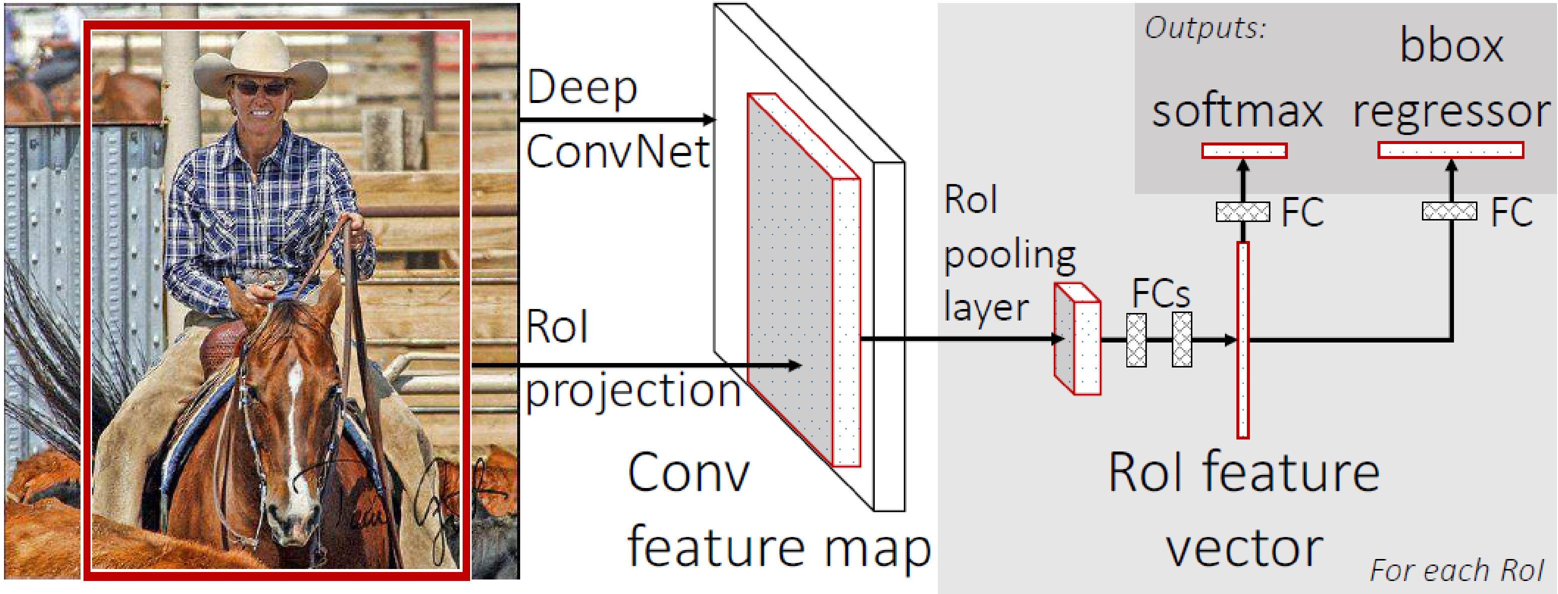


  - Arbitrary size image => fixed-length representation
  - Implemented by max-pooling operations
  - Speeds testing up

2. Region proposals + CNN

- Idea (2):
  - Refine bounding box by regression
  - Multi-task loss: classification + bounding box offset

- Fast R-CNN (= R-CNN + idea 1 + idea 2)
  - Girshick R. Fast R-CNN, ICCV 2015.



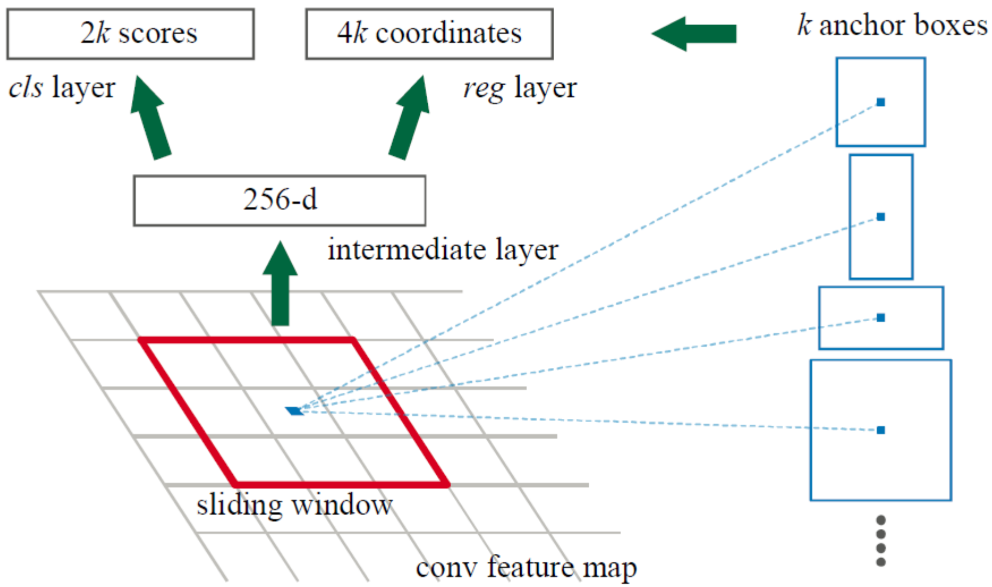
  - End-to-end training
  - Speed up both testing and training, but proposals still expensive

2. Region proposals + CNN

- Idea (3):
  - Implement region proposal mechanism by CNN with shared convolutional features

⇒ Faster R-CNN

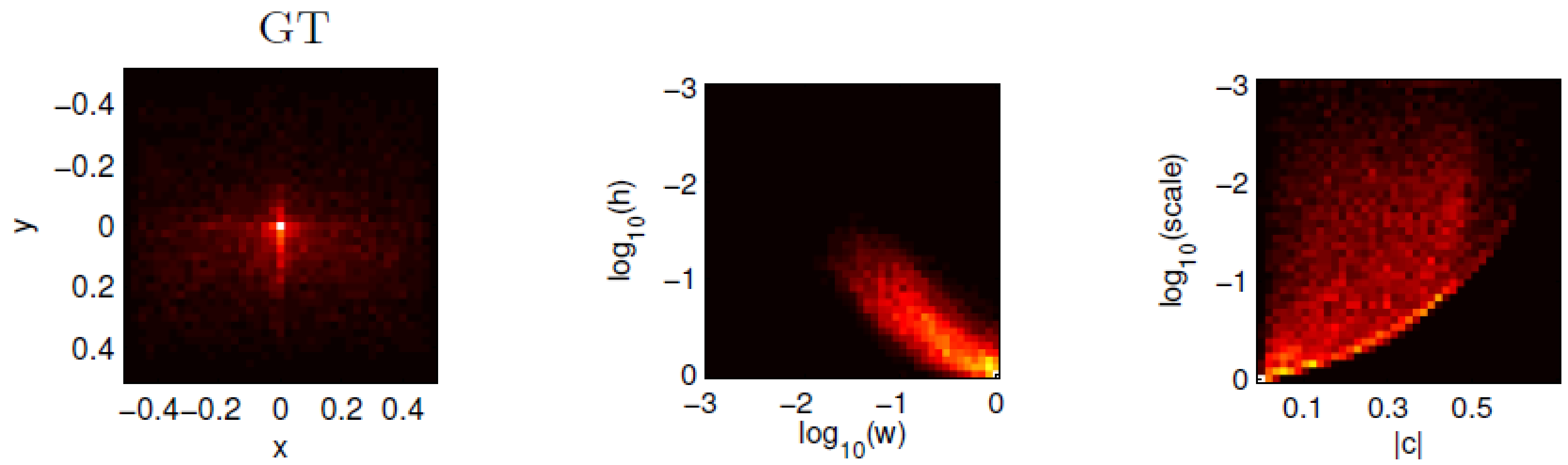
  - Ren et al. Faster R-CNN: *Towards Real-Time Object Detection with Region Proposal Networks*. NIPS 2015.
  - On top of the last **conv** layer **slide** a small region proposal network



  - Training: simple alternating optimization (RPN, fast R-CNN)
  - Accurate: 73.2% mAP (VOC 2007), Fast: 5 fps

3. Detection CNN without region proposals

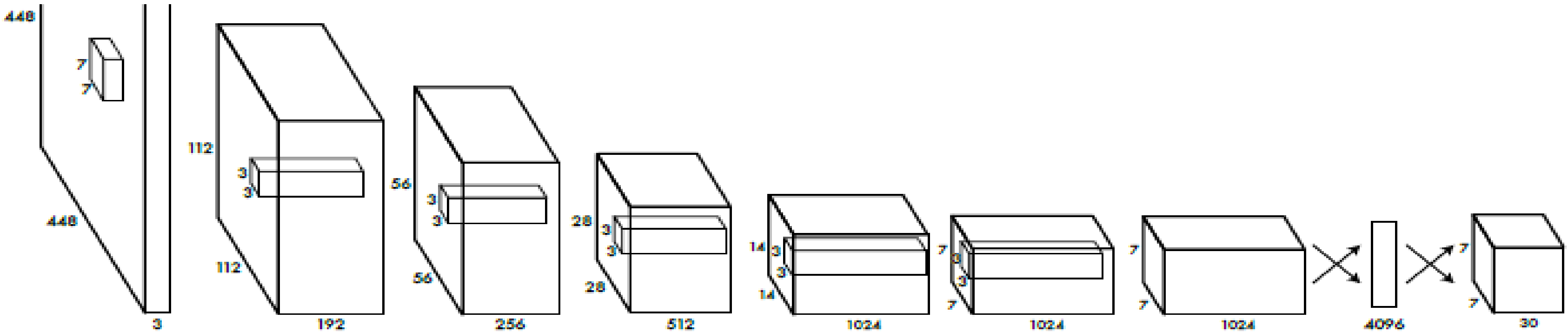
- R-CNN minus R
  - K. Lenc, A. Vedaldi. *R-CNN minus R*. BMVC 2015.
  - Sophisticated region proposal algorithm unnecessary, since a constant region coverage works well
    - Fixed 3k region proposals generated by K-means clustering of training data bounding boxes

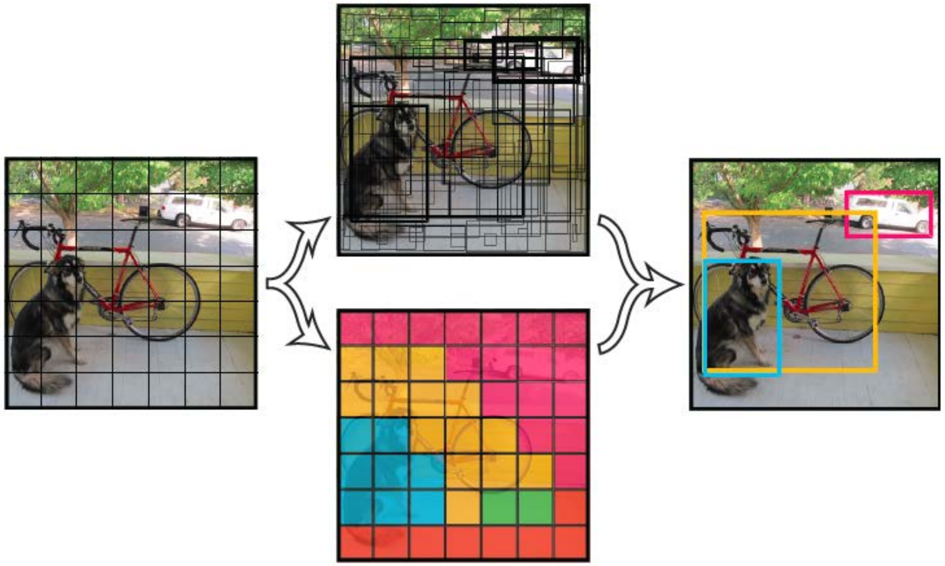


- Streamlined into a single network (fast training and testing)
- Competitive results despite simplicity (53% mAP, VOC 2007)
  - Can recover from imperfect alignment

3. Detection CNN without region proposals

- YOLO "You Only Look Once"
  - Redmond et al. *You Only Look Once: Unified, Real-Time Object Detection.* CVPR 2016.
  - A single net predicts bounding boxes and class probabilities directly from the entire image in one evaluation





Output layer:

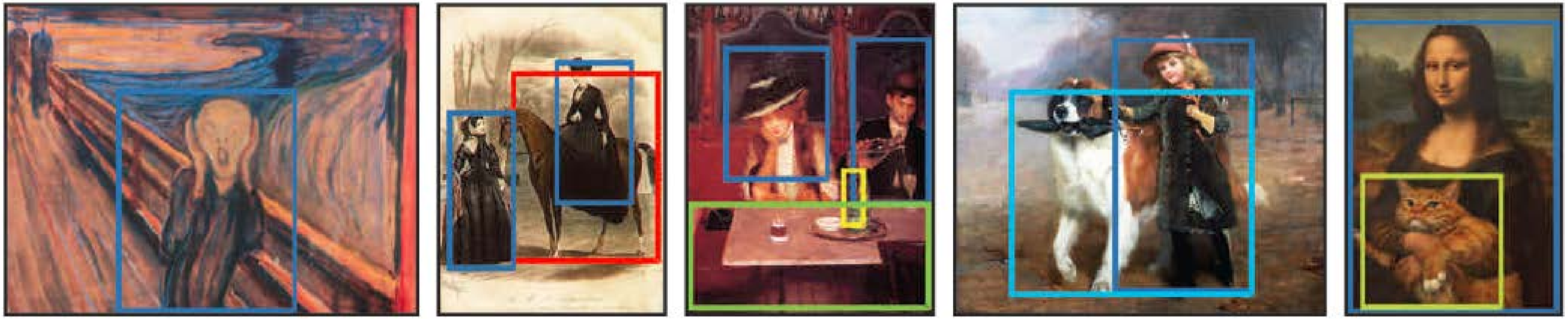
- Tensor 7x7x30

7x7 spatial grid
30=2*5+20

2: number of bboxes per cell
5: (x,y,w,h, overlap score)
20: number of classes

3. Detection CNN without region proposals

- YOLO properties:
  1. Reasons globally
     - Entire image is seen for training and testing, contextual information is preserved (=> less false positives)
  2. Generalization
     - Trained on photos, works on artworks

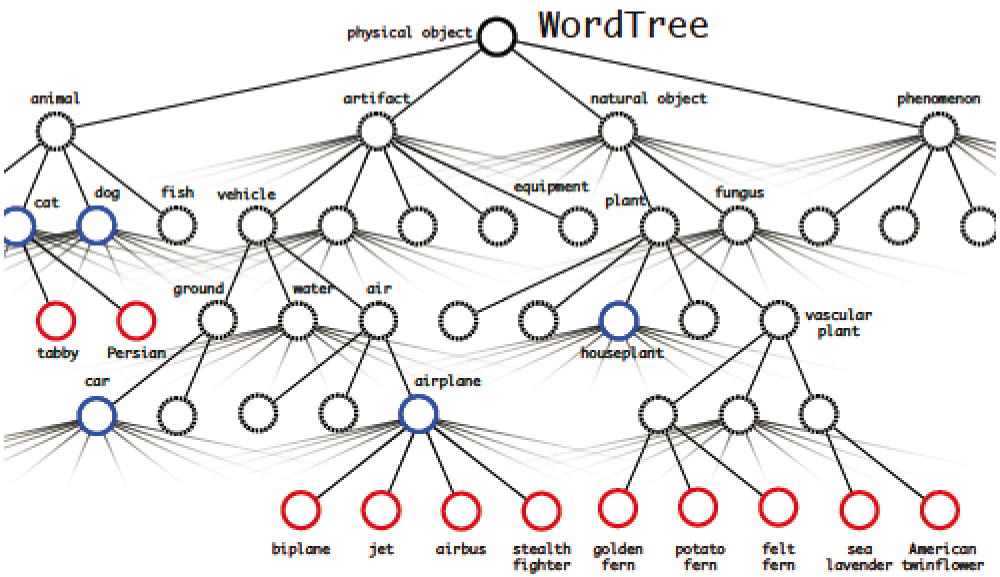


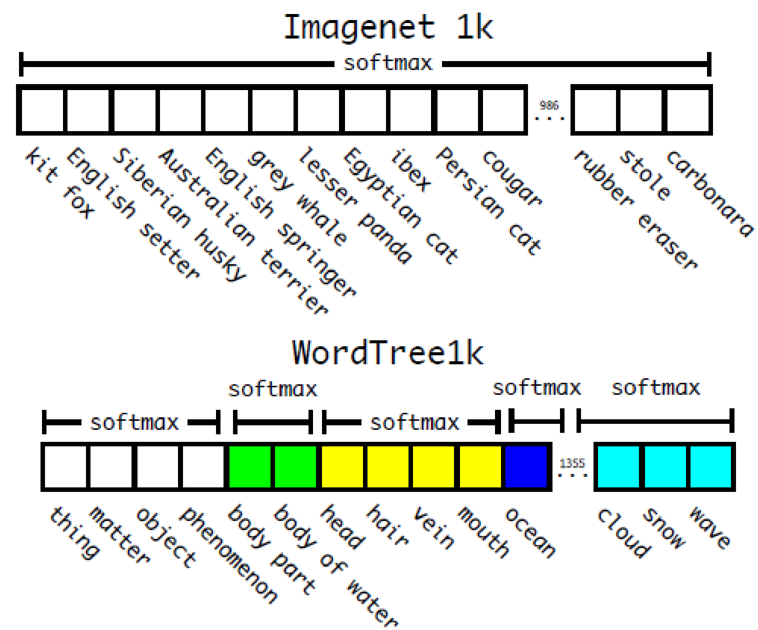
  3. Fast (real-time)

	mAP (VOC 2007)	FPS (GPU Titan X)
YOLO	63.4%	45
fast YOLO	52.7%	150

3. Detection CNN without region proposals

- YOLOv2, YOLO 9000
  - Redmon J., Farhadi A. *YOLO9000: Better, Faster, Stronger.* CVPR 2017 (To Appear)
  - Several technical improvements:
    - Batch normalization, Higher resolution input image (448x448), Finer output grid (13x13), Anchor boxes (found by K-means)
  - Hierarchical output labels:

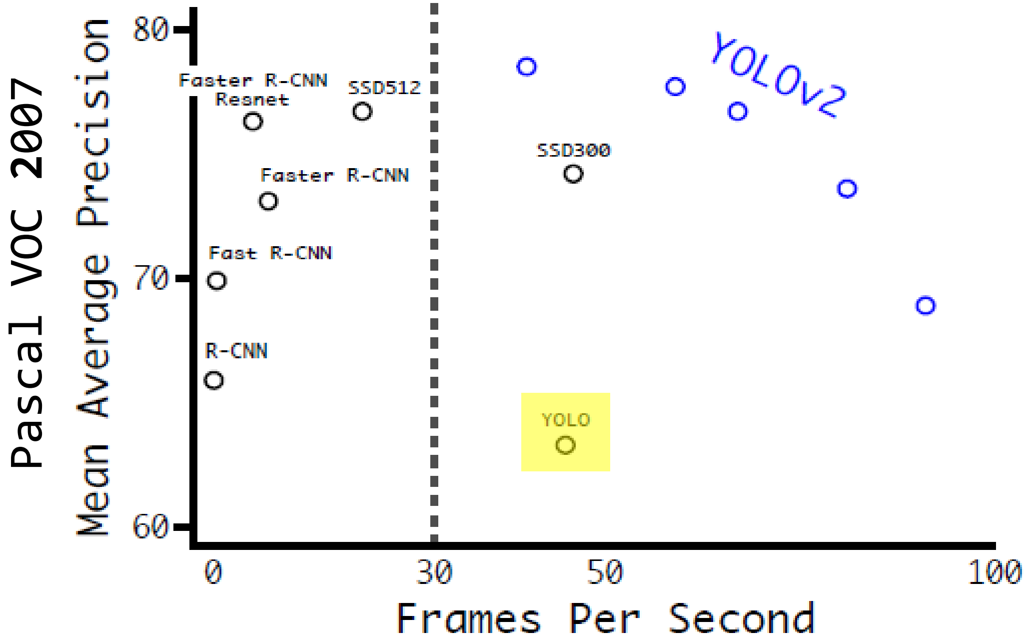




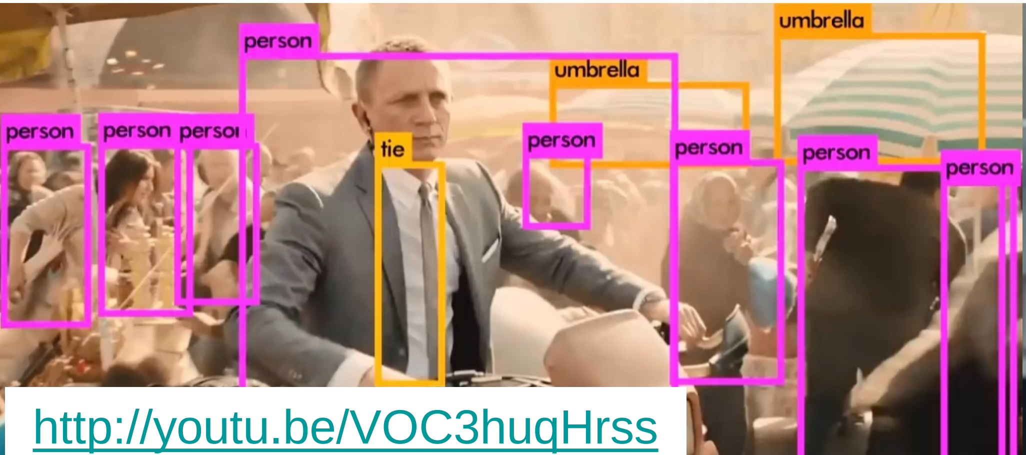
  - Able to learn from images without bounding box annotation (weak supervision)

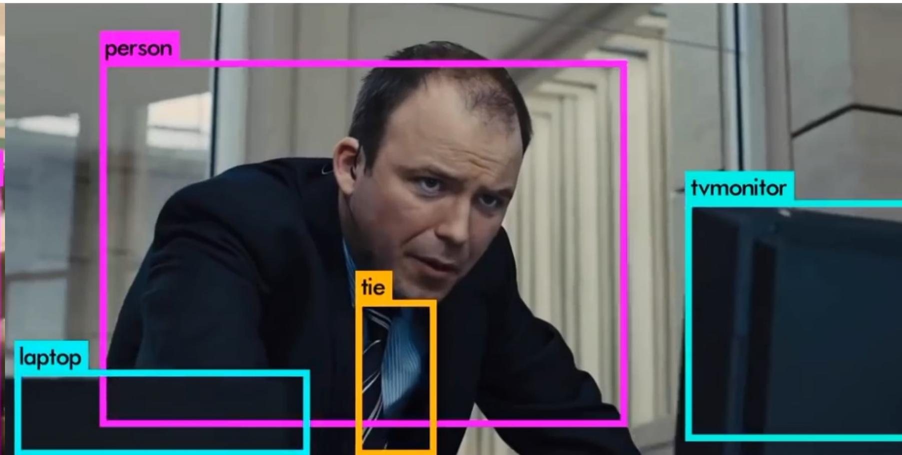
3. Detection CNN without region proposals

- YOLOv2, YOLO 9000 summary



  – The most accurate, the fastest… [video]





Detection CNN - summary

1. Exhaustive scanning windows + CNN

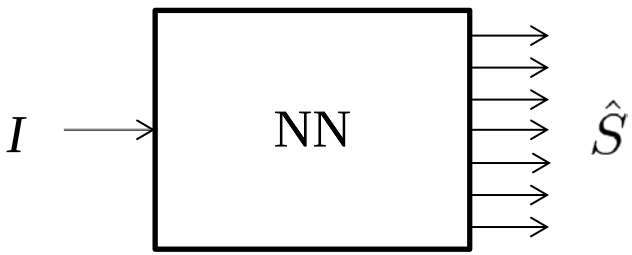
2. Region proposals + CNN
   1. R-CNN
   2. SPP net
   3. Fast R-CNN
   4. Faster R-CNN

3. CNN without region proposals
   1. R-CNN minus R
   2. YOLO
   3. YOLO v2, YOLO 9000

Some Insight into the Deep Nets

Deep Network Can Easily Be Fooled

- Szegedy et al. Intriguing properties of neural networks. ICLR 2014
  - Small perturbation of the input image changes the output of the trained “well-performing” neural network
  - The perturbation is a non-random image, imperceptible for human

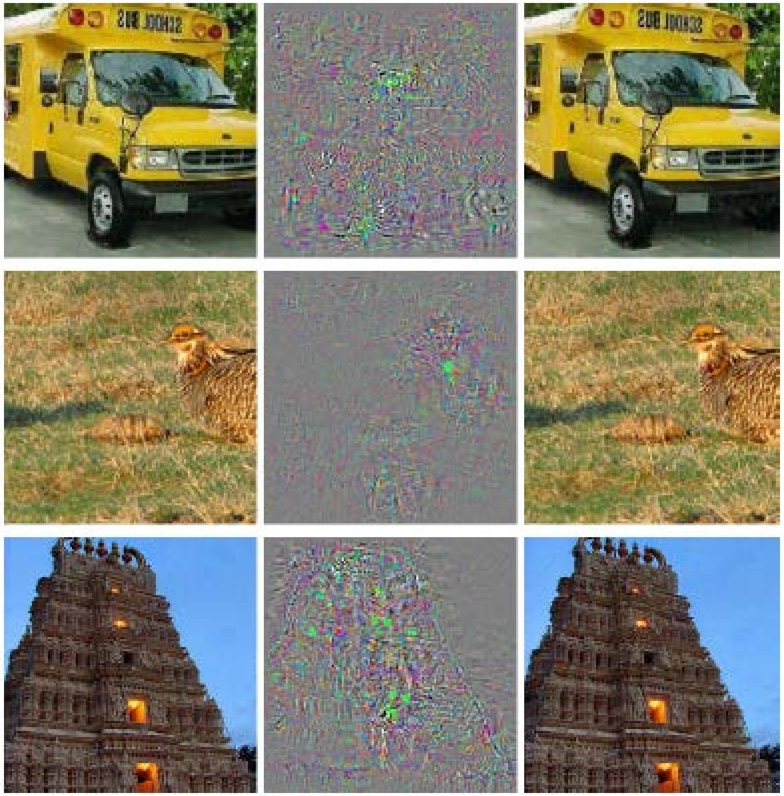


$$\min_r \{ ||\mathrm{NN}(I + r) - S||^2 + \lambda ||r||^2 \}$$

  - Optimum found by gradient descent

$$r^{t+1} = r^t - \gamma\left(\frac{\partial \mathrm{NN}(I)}{\partial I} + 2\lambda r\right)$$

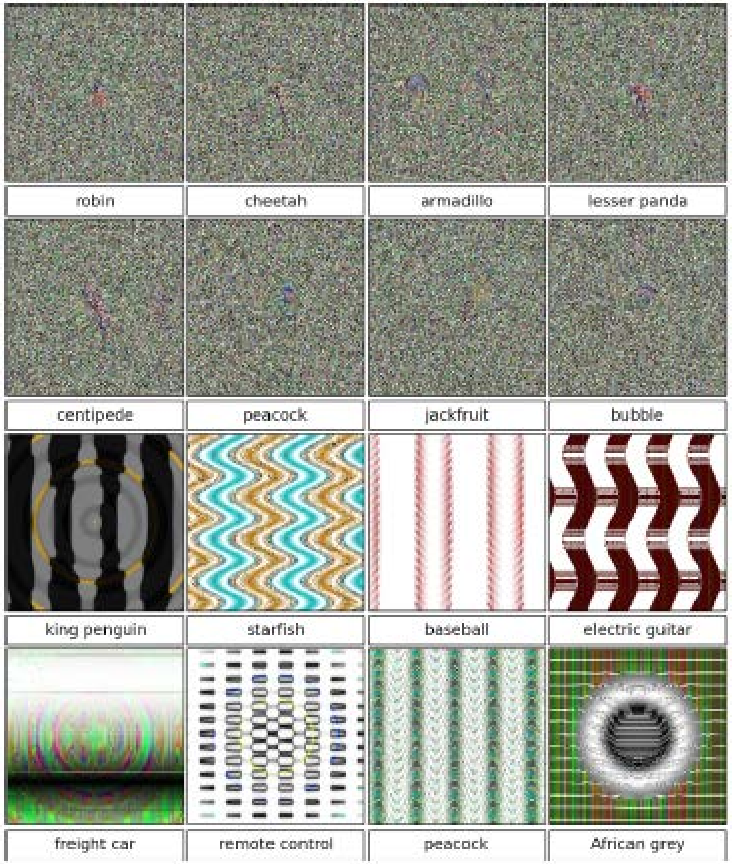
ostrich



Deep Network Can Easily Be Fooled

- Nguyen et al. *Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images*. CVPR 2015.
  - Artificial images that are unrecognizable to humans, producing high output score can be found
  - The optimum images found by evolutionary algorithm
    - Starting from random noise
    - Direct/Indirect encoding

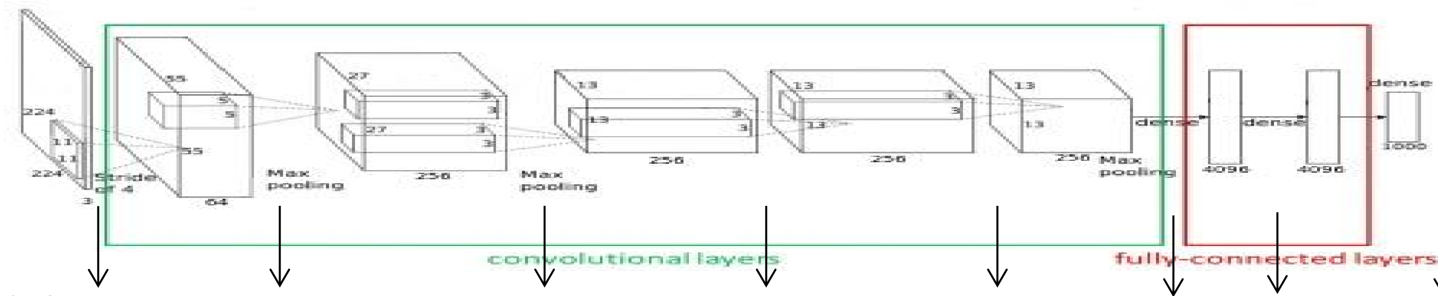
$$\min_I ||\mathrm{NN}(I) - S||^2$$



$\Rightarrow$ The images found do not have the natural image statistics

Visualization the Deep Nets

- Mahendran A., Vedaldi A. *Understanding Deep Image Representations by Inverting Them.* CVPR 2015.



$\Phi_1(I) = I$ $\Phi_k(I)$ $\Phi_K(I) = S$

- Include image regularization (natural image prior)

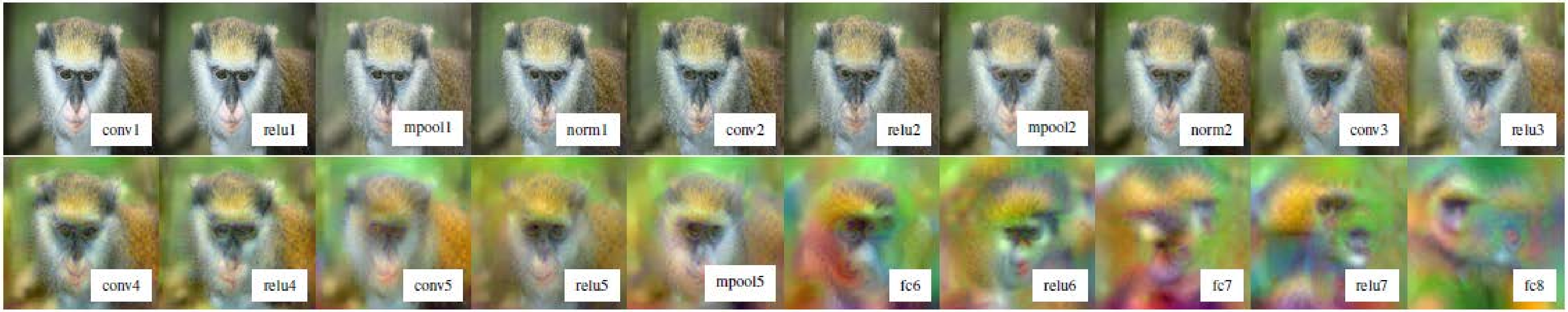
$$\min_I \{ ||\Phi_k(I) - \Phi_k^0||^2 + \lambda R(I) \}$$

- Total Variation regularizer (TV)

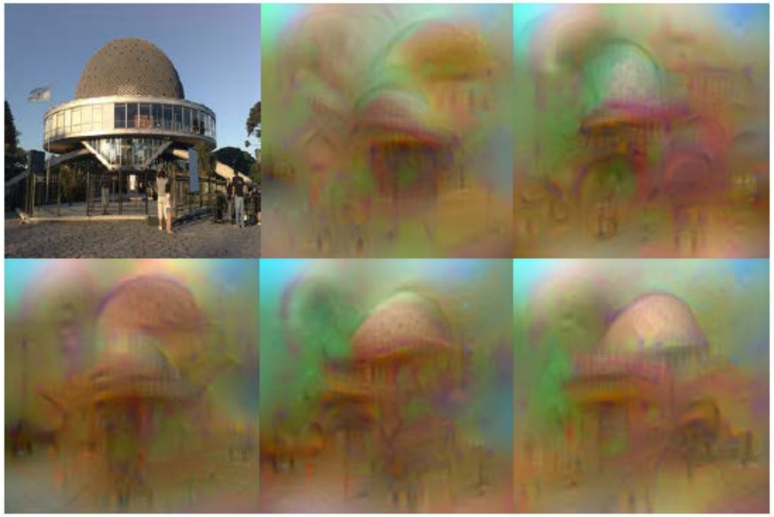
$$R(I) = \sum_{x,y} \Big((\frac{\partial I(x,y)}{\partial x})^2 + (\frac{\partial I(x,y)}{\partial y})^2 \Big)^{\frac{\beta}{2}}$$

Visualizing the Deep Nets

- CNN reconstruction



  – Gradient descent from random initialization
  – Reconstruction is not unique



  ⇒ All these images are identical for the CNN

- Similarly, find an image that causes a particular neuron fires (maximally activate)

Deep Dream

- Start from an original image
- Manipulate the input image so that response scores are higher for all classes
- Regularization with TV prior

[video]

http://youtu.be/EjiyYtQIEpA

Deep Dream

- Maybe…

Hieronymus Bosch, Garden of Earthly Delights (~1510), [part]

Salvador Dalí

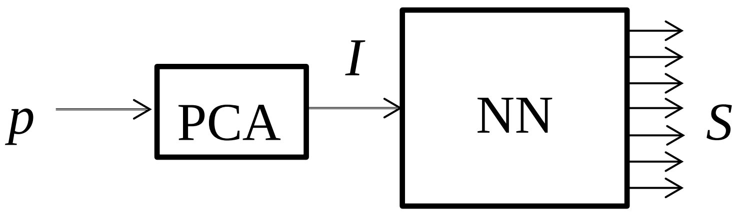
Soft Construction with Boiled Beans (1936)

Swans Reflecting Elephants (1937)

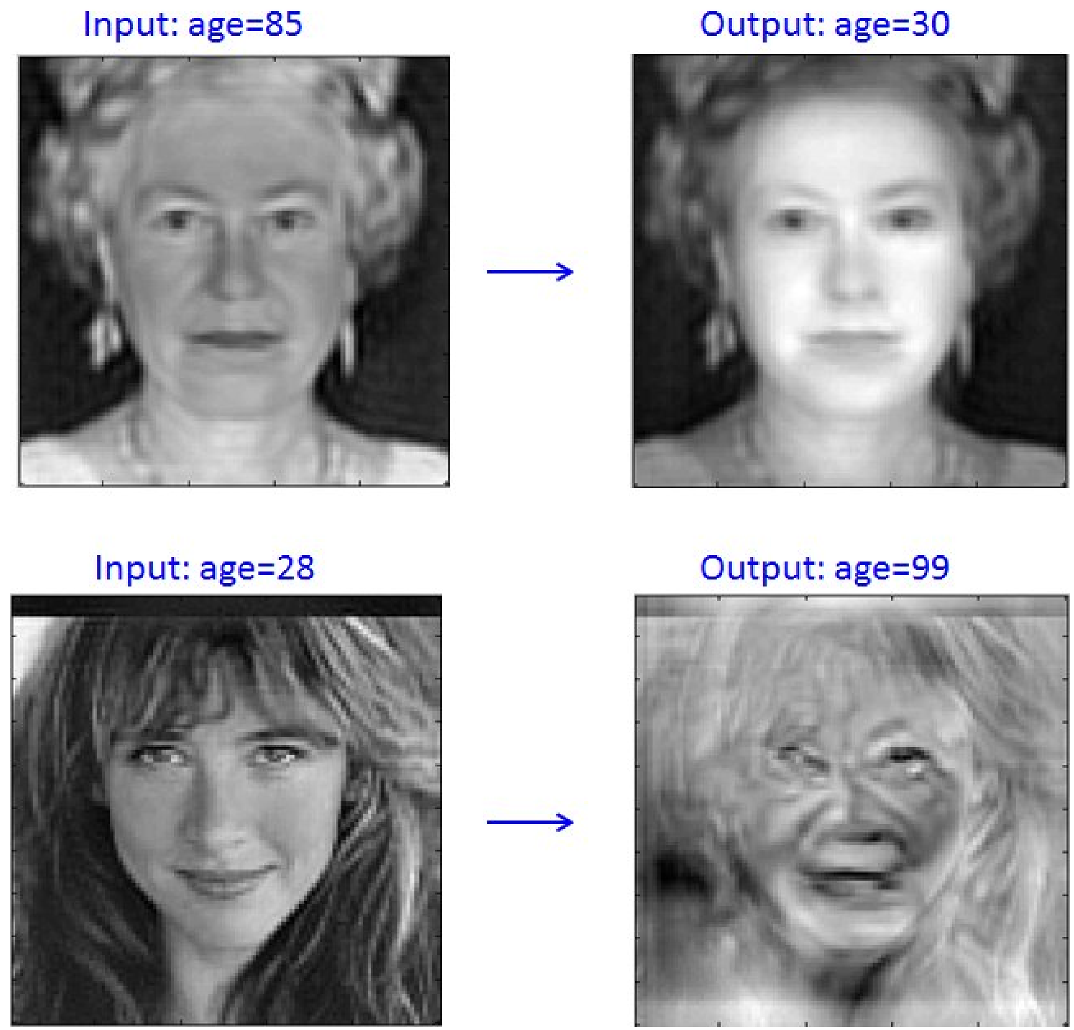
Apparition of a Face and Fruit Dish on a Beach (1937)

Deep Aging

- Our network trained for predicting age (gender and landmarks) was used



$$\min_{p} ||\mathrm{NN}(\mathrm{PCA}(p)) - S^t||^2$$



[Čech, J. Unpublished experiment]

Deep Art – Neural Style

- Gatys et al. *A Neural Algorithm of Artistic Style.* Journal of Vision, 2015.
  - Generate high-quality artistic rendering images from photographs
  - Combines content of the input image with a style of another image

Content image

Style images

Result images

  - More examples at Deepart.io

Deep Art – Neural Style

- Main idea:
  - the style is captured by correlation of lower network layer responses
  - the content is captured by higher level responses

- The optimization problem:

$$\min_I \{\alpha L_{\text{content}}(I_1, I) + \beta L_{\text{style}}(I_2, I)\}$$

$$L_{\text{content}} = \sum_k ||\Phi_k(I) - \Phi_k(I_1)||^2$$

$$L_{\text{style}} = \sum_k w_k ||G(\Phi_k(I)) - G(\Phi_k(I_2))||^2$$

G is a Gram matrix (dot product matrix of vectorized filter responses)

Summary

- Using Network gradient according to the image for various optimization
  - Fooling the net
  - Visualization
  - Dreaming, Hallucination
  - Aging
  - Artistic rendering of photographs

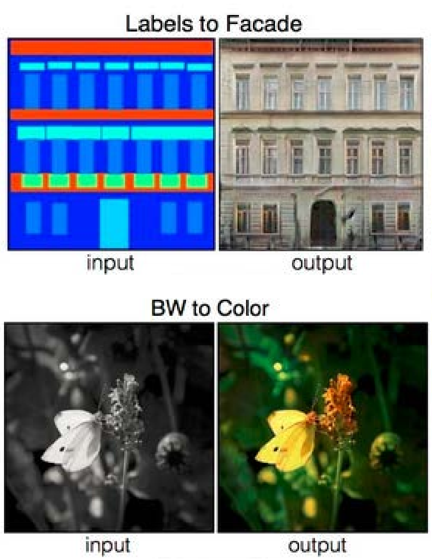
=> Understanding of the trained model

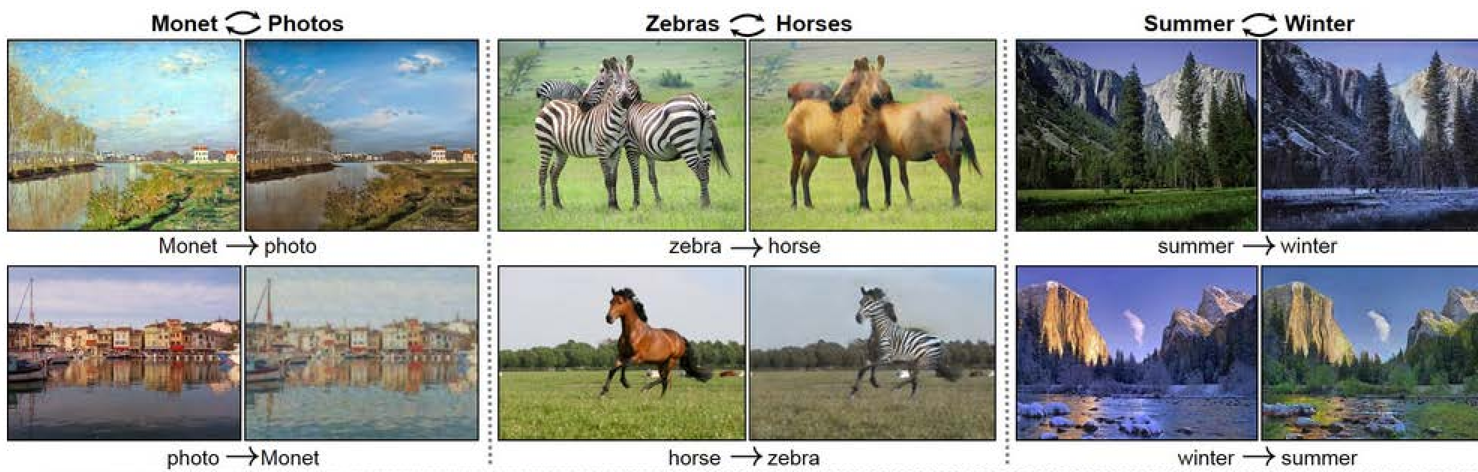
What was not mentioned…

What was not mentioned…

- Generative Adversarial Networks (GANs)
  - Generative models, samples are almost photo-realistic

  - Image to image translation [Efros et al. @ Berkeley]





What was not mentioned…

- Recurrent NNs
  - Processing sequences, prediction, image captioning, etc.
  - Fei-Fei Li, Andrej Karpathy @ Stanford

Shakespeare

VIOLA: Why, Salisbury must find his flesh and thoughtThat which I am not aps, not a man and in fire, To show the reining of the raven and the wars To grace my hand reproach within, and not a fair are hand, That Caesar and my goodly father's world; When I was heaven of presence and our fleets, We spare with hours, but cut thy council I am great, Murdered and by thy master's ready there My power to give thee but so much as hell: Some service in the noble bondman here, Would show him to her wine.

KING LEAR: O, if you were a feeble sight, the courtesy of your law, Your sight and several breath, will wear the gods With his heads, and my hands are wonder'd at the deeds, So drop upon your lordship's head, and your opinion Shall be against your honour.

LaTeX algebra

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*
Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

"little girl is eating piece of cake."

Linux source code

```c
/*
 * Increment the size file of the new incorrect UI_FILTER group infor
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
```

"a young boy is holding a baseball bat."

Conclusions

- No doubt that the paradigm is shifting/has shifted
- Turbulent period
  - The research is extremely accelerated, many novel approaches
  - New results are still astonishing
- …

- Isn't it all fascinating?