Deep Convolutional Neural Networks II.

Jan Čech

Lecture Outline

1. Deep convolutional networks for Object detection
2. Deep convolutional networks for Semantic segmentation
3. “Deeper” insight into the Deep Nets
4. Generative Models (GANs)
5. What was not mentioned…

Deep Convolutional Networks for Object Detection

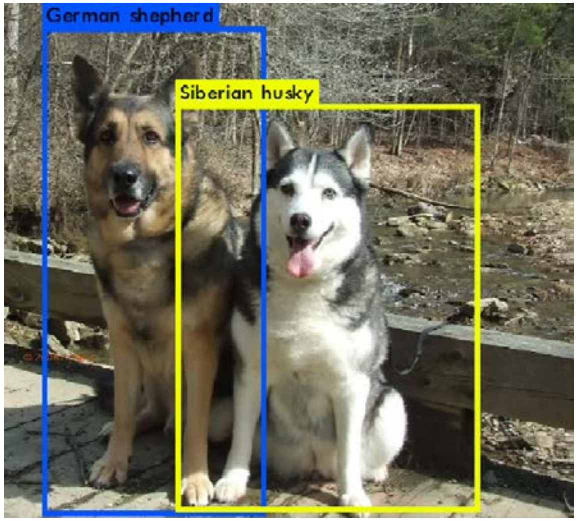
Convolutional Networks for Object Detection

- What is the object detection?

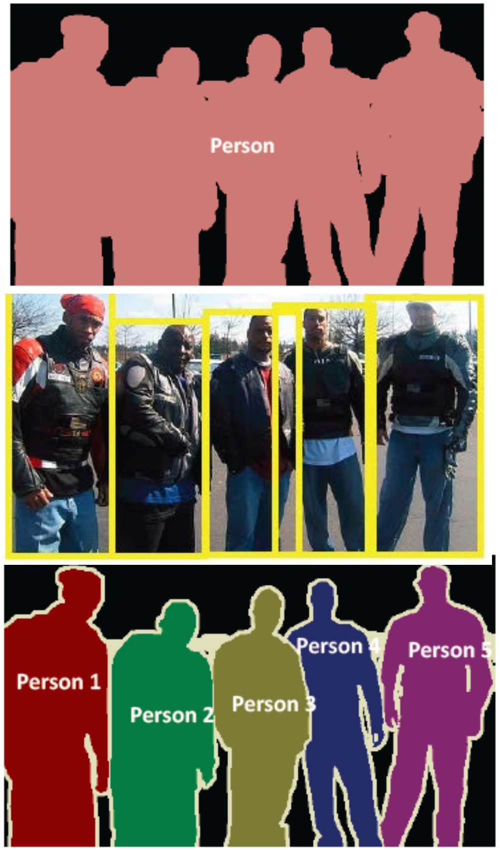
Semantic segmentation
- What + Where?
- Pixel-level accuracy

Grocery store

Image recognition
- What?
- holistic



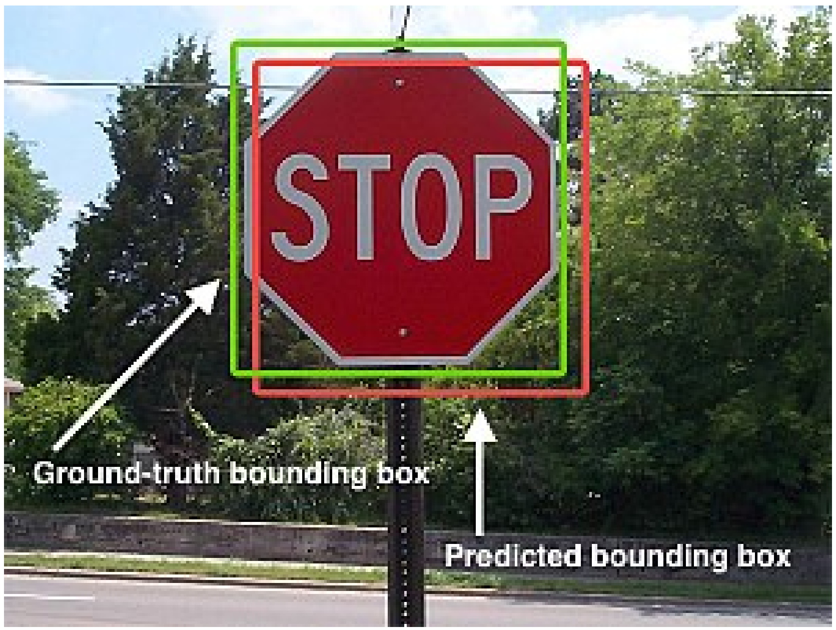
Object detection
- What + Where?
- Bounding boxes



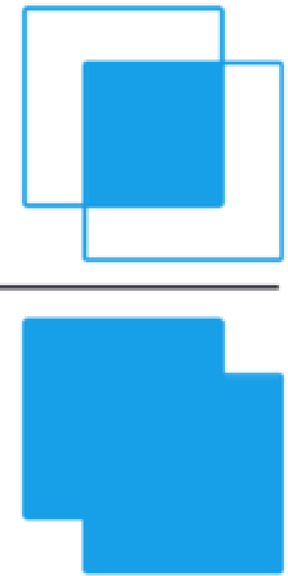
Instance segmentation
- What instance + Where
- Pixel-level accuracy

How to measure detector accuracy?

- Ground-Truth bounding boxes, Detections – predicted bounding boxes
- Intersection over Union (IoU), a.k.a. Jaccard index



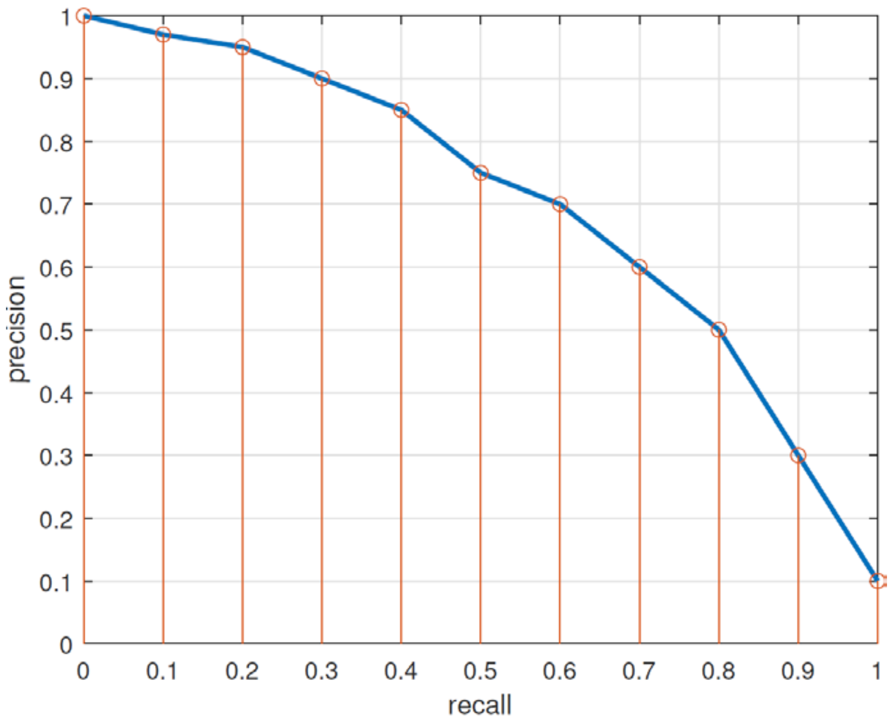
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

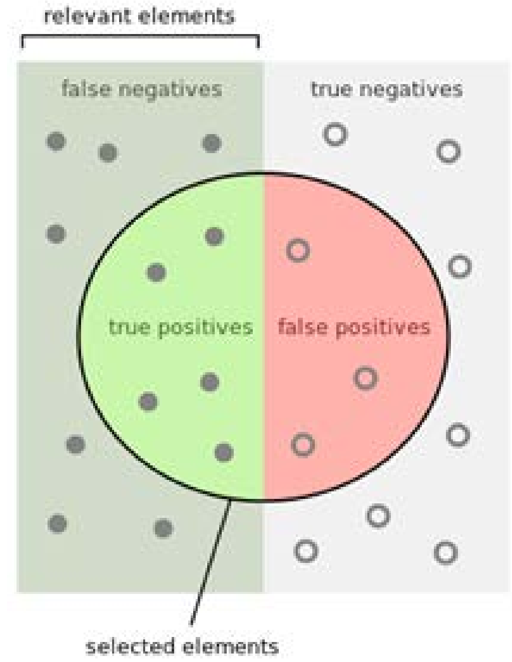


- A detection is correct (= true positive) if it has enough overlap with the ground-truth
  - Typically, IoU > 50%

How to measure detector accuracy?

- Mean Average Precision (mAP)





True positive: IoU > 50%

– Average Precision (Area under the precision-recall curve)

$$\mathrm{AP} = \int_r p(r)dr \approx \frac{1}{N}\sum_i p(r_i)$$

– Mean over all classes

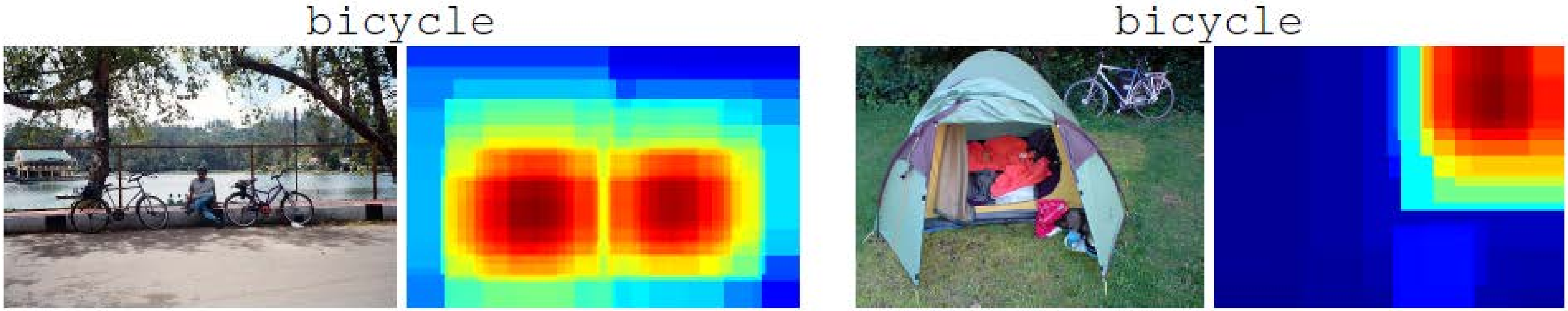
$$\mathrm{mAP} = \frac{1}{C}\sum_c \mathrm{AP}_c$$

Pascal VOC 2007 challenge
(N = 11, r = 0:0.1:1)
(C = 20)
Classes: Person, bird, cat, car, ...

1. Scanning window + CNN

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

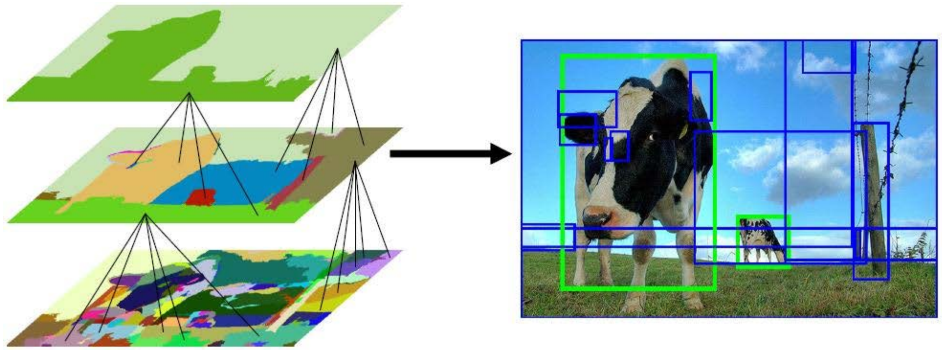
- A trivial detection extension - exhaustive scanning window
  1. Scan all possible bounding boxes
  2. Crop bounding box, warp to 224x224 (fixed-size input image)
  3. Run CNN
- Works, but
  - prohibitively slow…



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

2. Region proposals + CNN

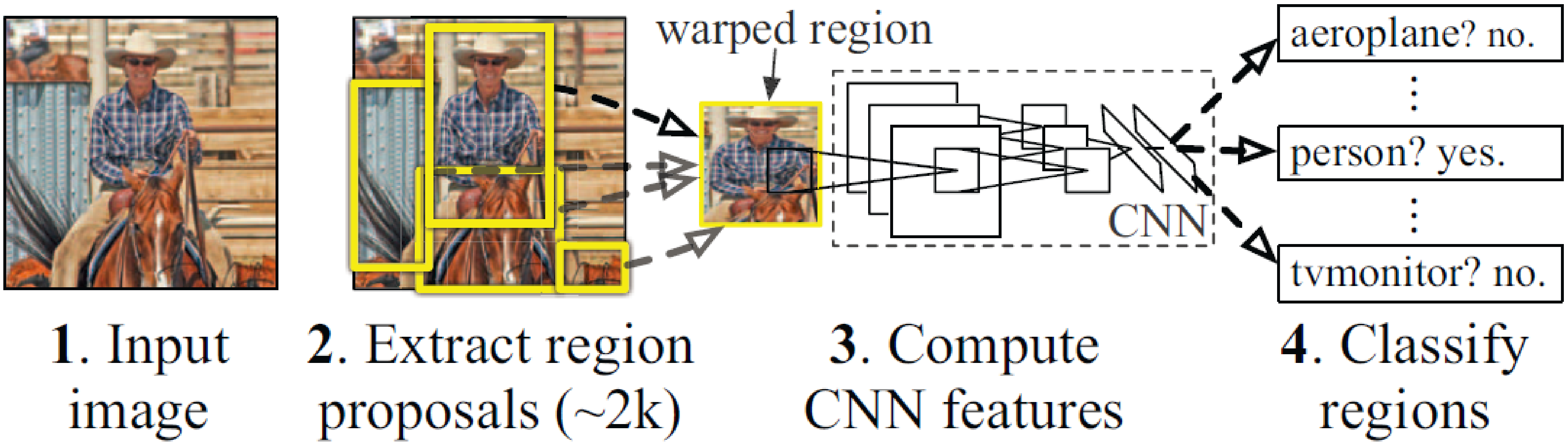
- CNN not evaluated exhaustively, but on regions where objects are likely to be present
- Region proposals (category independent):
  - Selective search [Uijlings-IJCV-2013]



  - Edgeboxes [Zitnick-ECCV-2014]

2. Region proposals + CNN

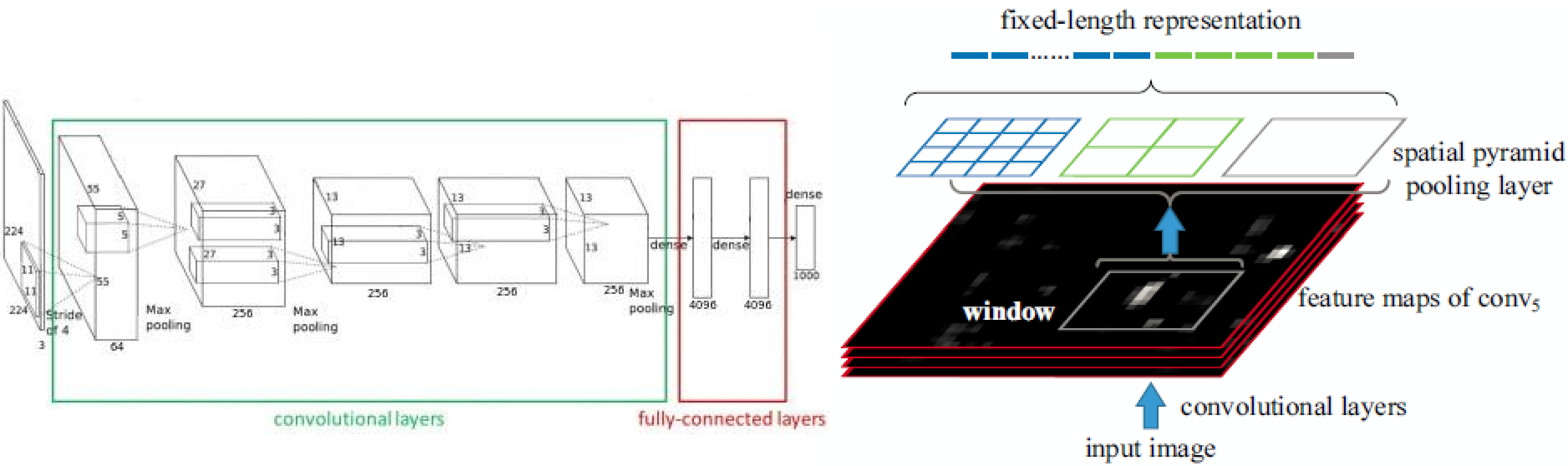
- R-CNN “Regions with CNN feature”
  - Girshick et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. CVPR 2014.



- 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.

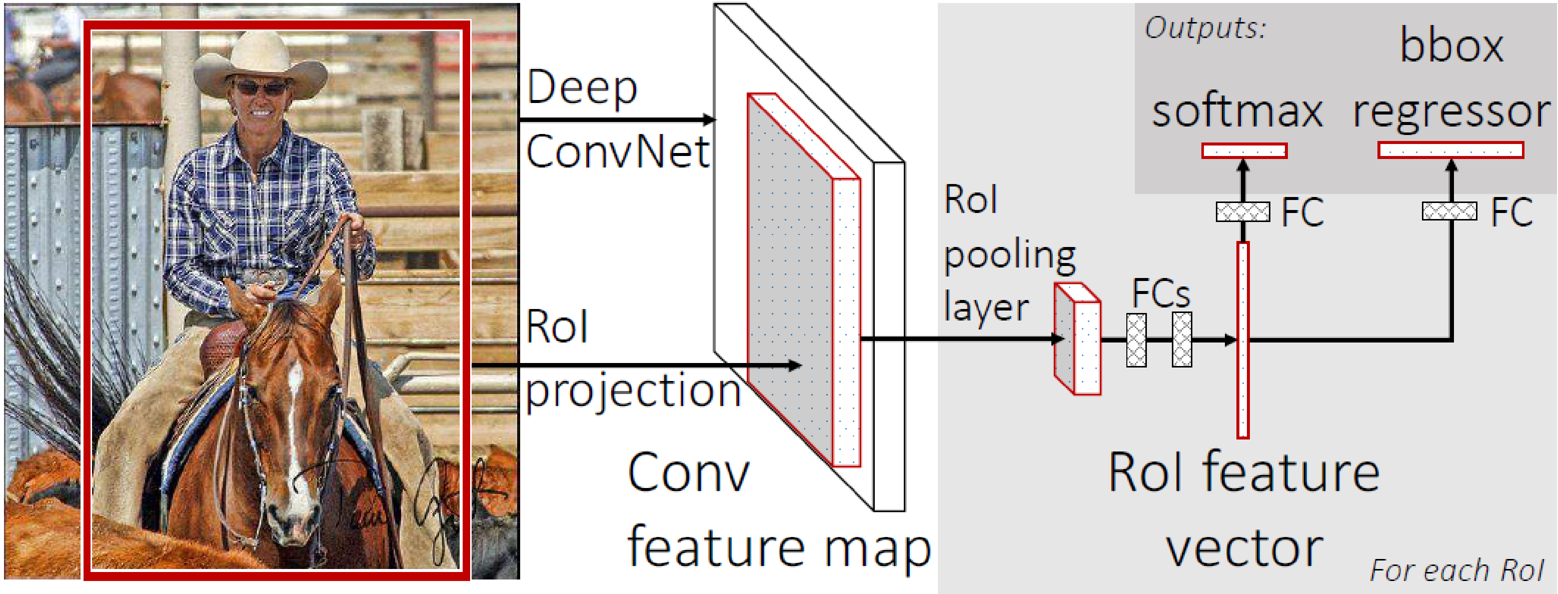


  - Arbitrary size image => fixed-length representation
  - Implemented by max-pooling operations
  - Speeds testing up

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.

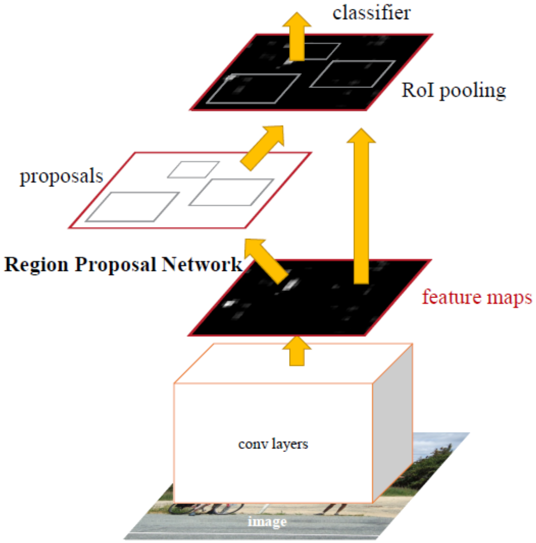


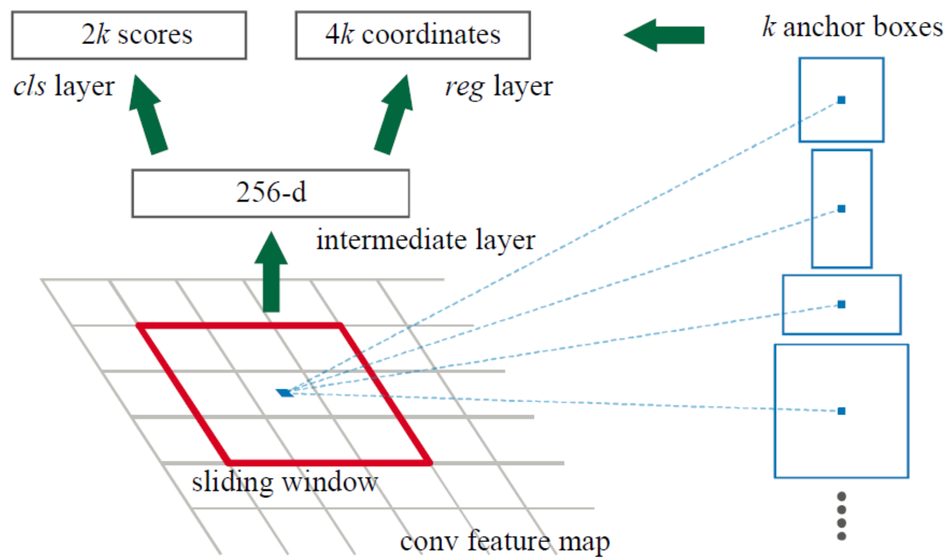
  - End-to-end training
  - Speed up, but proposals still expensive

2. Region proposals + CNN

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

⇒ Faster R-CNN
  - Ren et al. [Faster R-CNN: *Towards Real-Time Object Detection with Region Proposal Networks*](). NIPS 2015.
  - Region proposal network: object/not-object + bb coord. (k-anchor boxes)

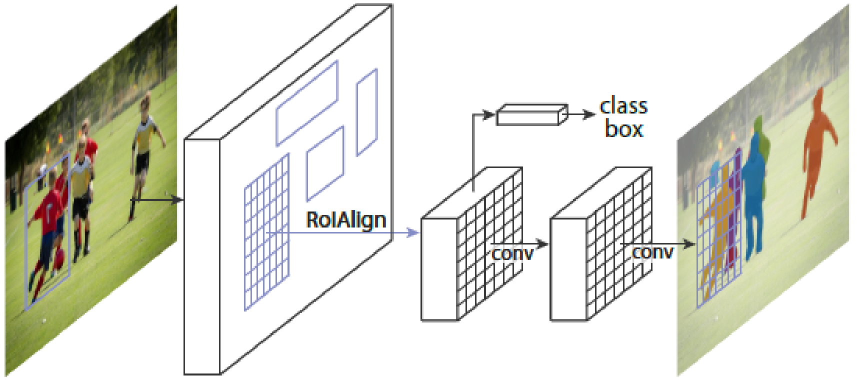


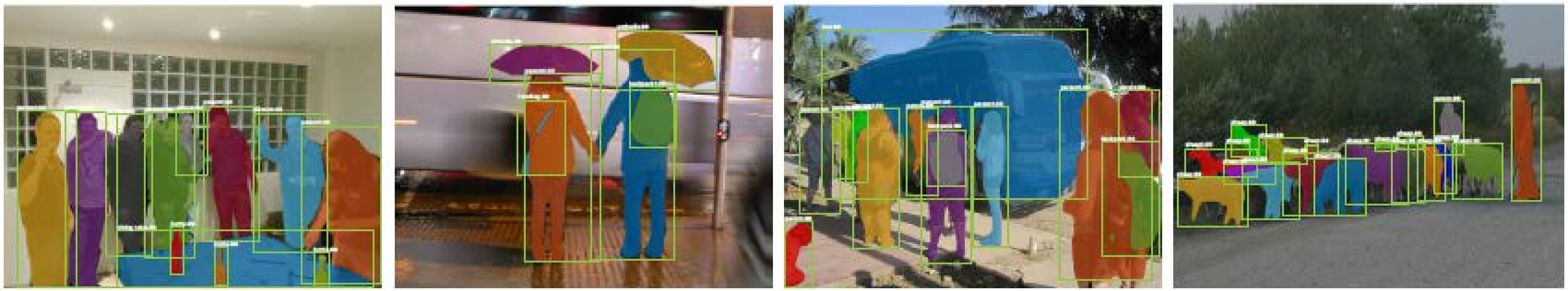


  - Training: simple alternating optimization (RPN, fast R-CNN)
  - Accurate: 73.2% mAP (VOC 2007), Fast: 5 fps

2. Region proposals + CNN + Instance segmentation

- Mask R-CNN
  - He et al., Mask R-CNN. ICCV 2017
  - Faster R-CNN + fully convolutional branch for segmentation
  - ROI alignment
    - Improved pooling with interpolation
  - Running 5 fps





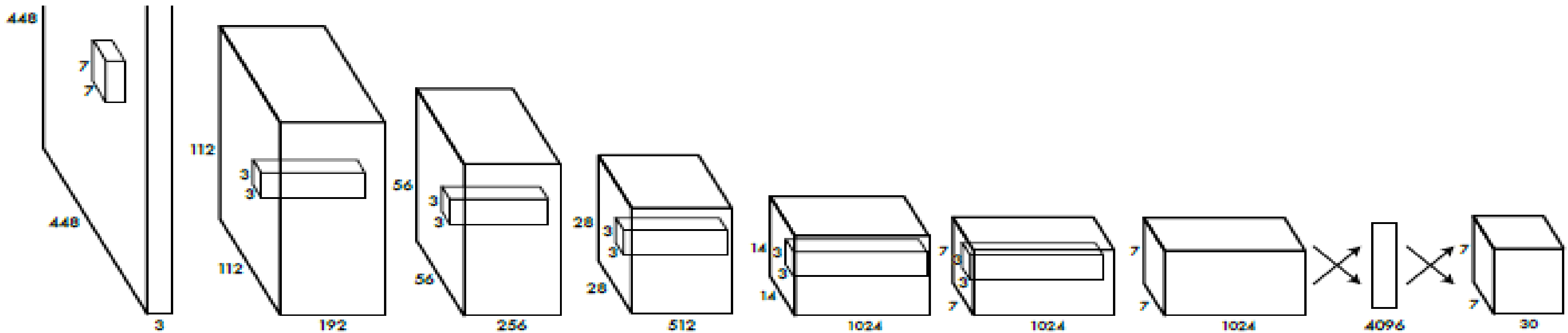
COCO dataset “Common Object in Context” (>200K images, 91 categories)

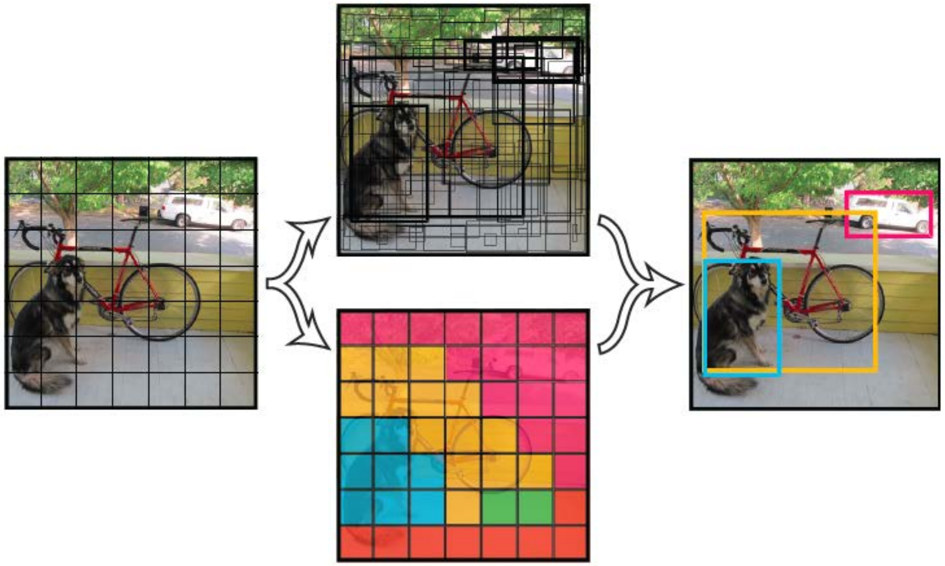
+ keypoint localization (pose estimation)

[video1] [video2]

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 a single execution





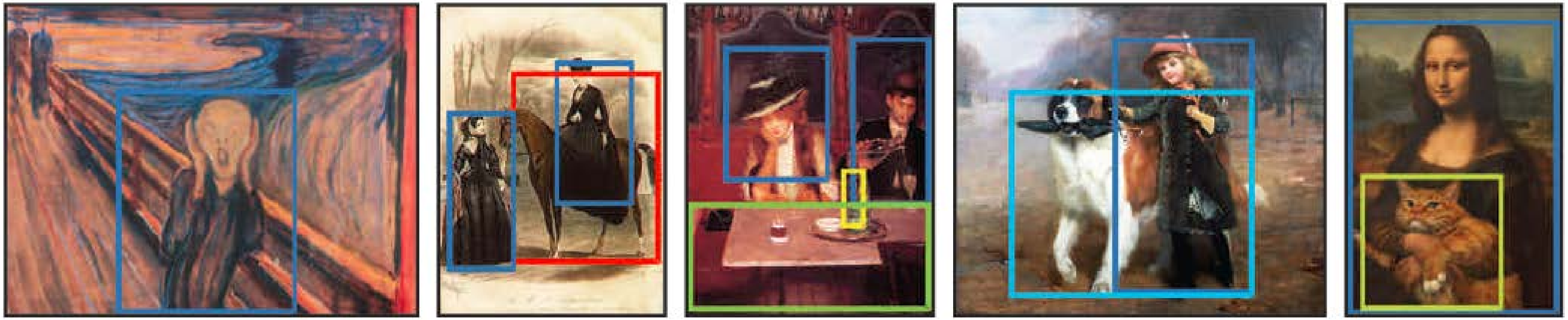
Output layer:
- Tensor 7x7x30

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

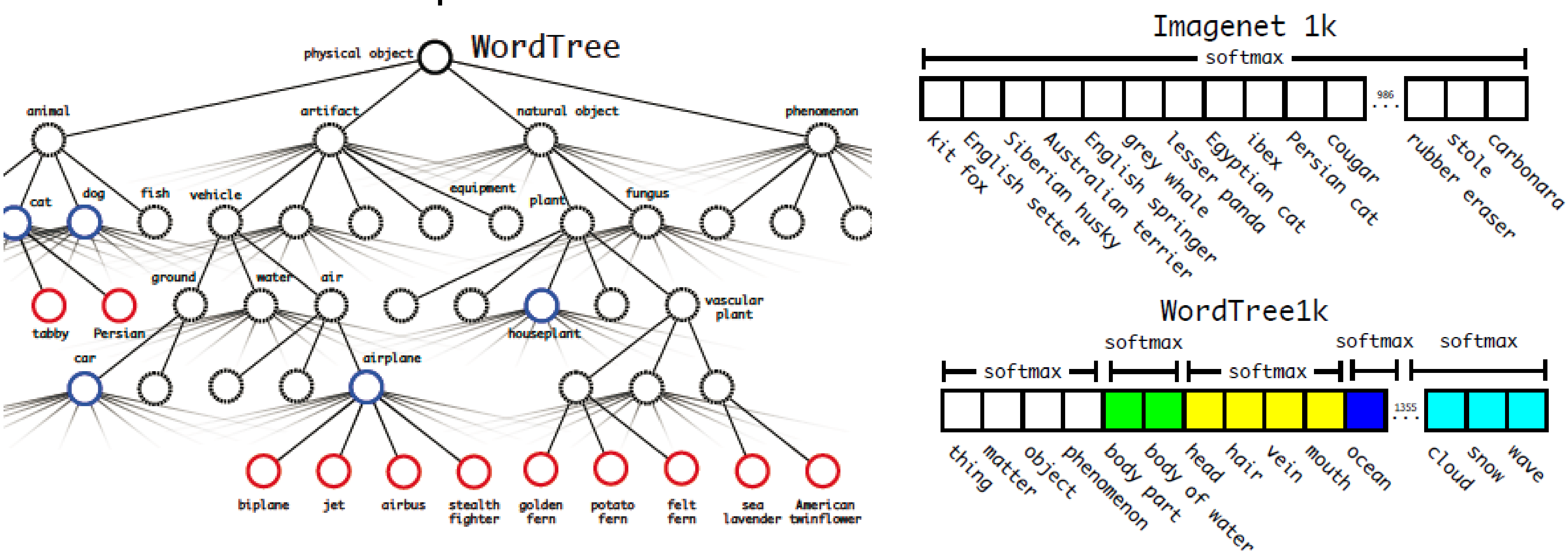


  3. Fast (real-time)

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

3. Detection CNN without region proposals

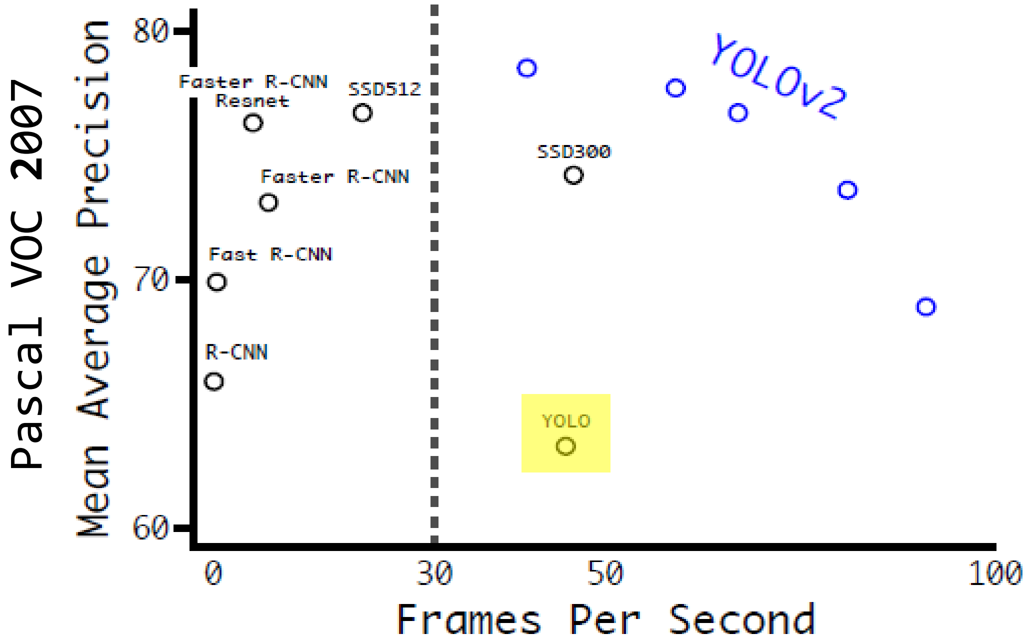
- YOLOv2, YOLO 9000
  - Redmon J., Farhadi A. [*YOLO9000: Better, Faster, Stronger*](). CVPR 2017
  - Several technical improvements:
    - Batch normalization, Higher resolution input image (448x448), Finer output grid (13x13), Anchor boxes (found by K-means)
  - Hierarchical output labels:



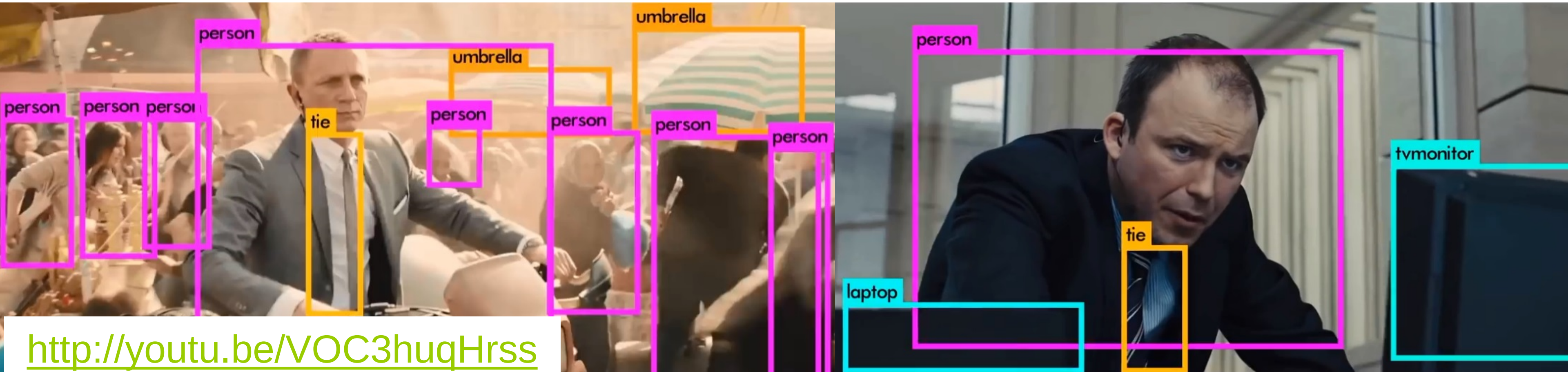
  - Trained on COCO and ImageNET datasets
  - Able to learn from images without bounding box annotation (weak supervision)

3. Detection CNN without region proposals

- YOLOv2, YOLO 9000 summary



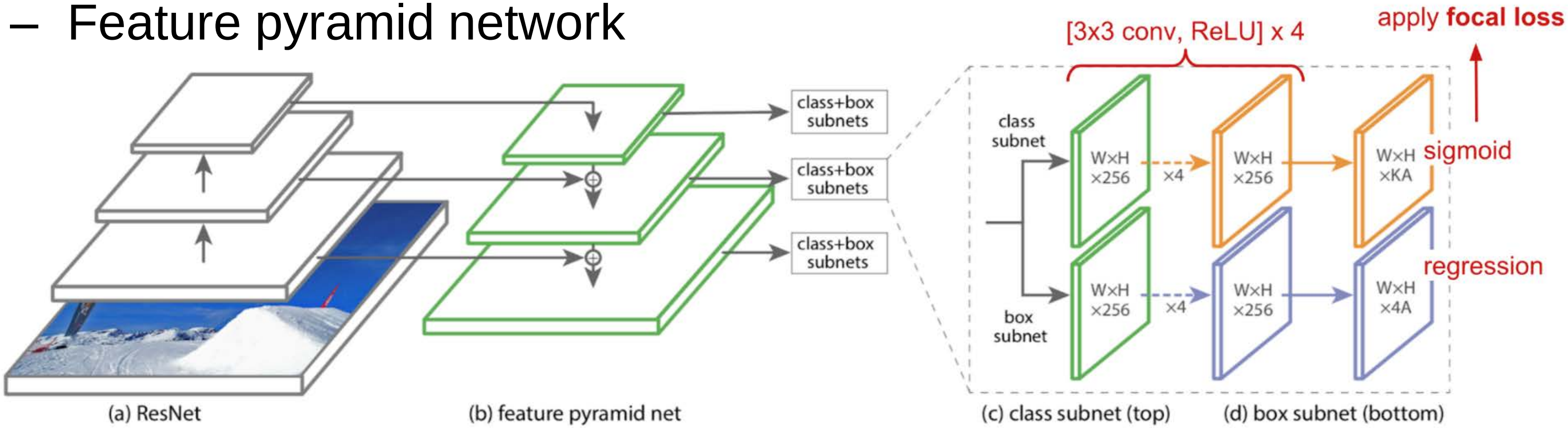
– The most accurate, the fastest… [video]



http://youtu.be/VOC3huqHrss

3. Detection CNN without region proposals

- RetinaNet (Lin et al., ICCV-2017, IEEE TPAMI 2020)
  - Feature pyramid network

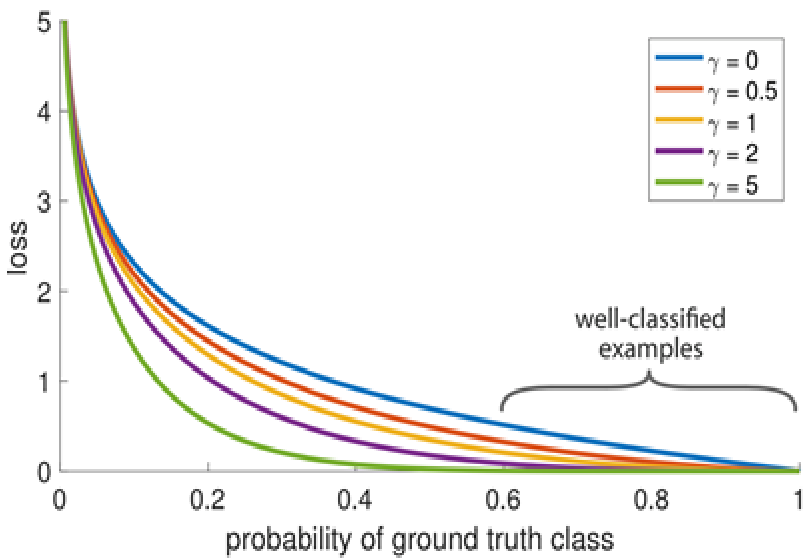


  - Focal Loss
    - Imbalance between positive and negative (background) classes (1:1000)
    - Assign more weight on hard examples

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise,} \end{cases}$$

$$\mathrm{CE}(p_t) = -\log(p_t)$$ Cross-entropy loss

$$\mathrm{FL}(p_t) = -(1 - p_t)^\gamma \log(p_t)$$ Focal loss



Detection CNN - summary

1. Exhaustive scanning windows + CNN

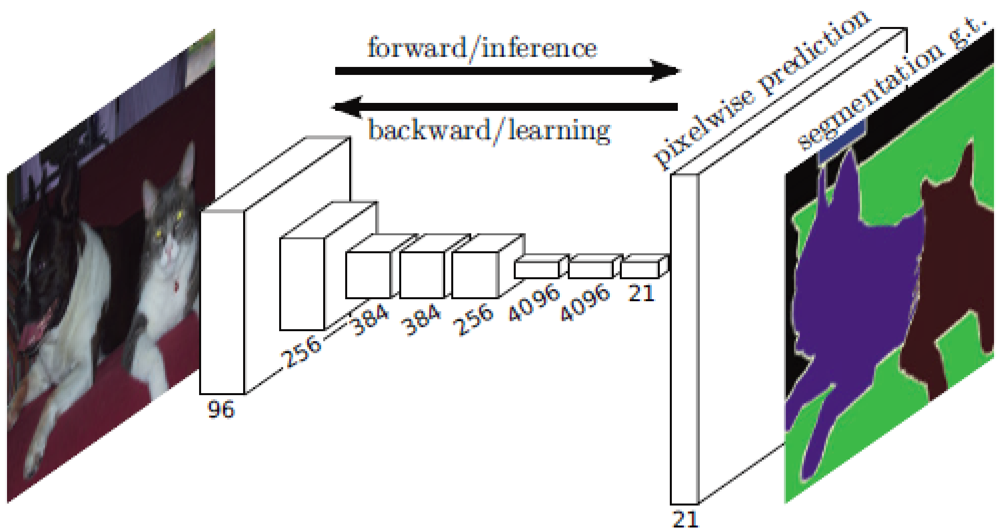
2. Region proposals + CNN
    1. R-CNN
    2. Fast R-CNN
    3. Faster R-CNN
    4. Mask R-CNN

3. CNN without region proposals
    1. YOLO
    2. YOLO v2, YOLO 9000
    3. RetinaNet

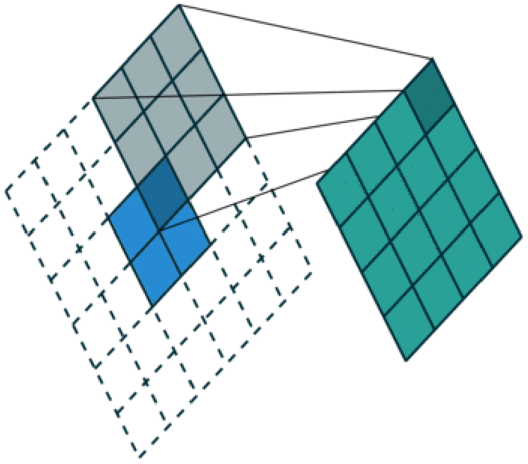
Deep Convolutional Networks for Semantic Segmentation

Fully Convolutional Net (FCN)

- Shelhammer et al. Fully Convolutional Networks for Semantic Segmentation, TPAMI 2017 (originally CVPR, 2015)

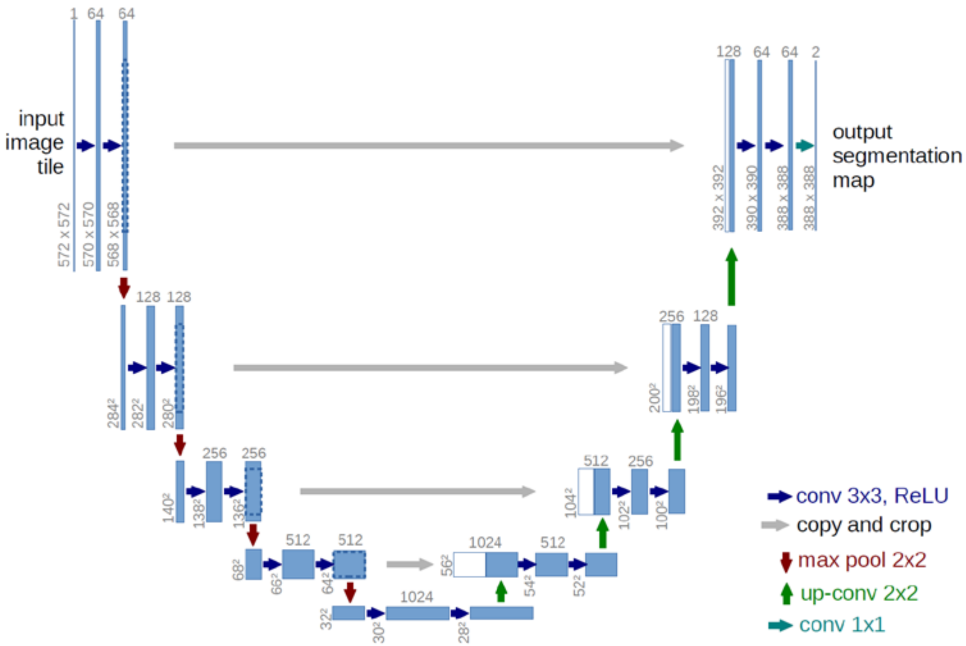


- Fully Convolutional (no fully connected layers)
  – The output size proportional to input size
- Upsamling at the last layer
  – Deconvolution layer (= transposed convolution, fractional-strided convolution)
  – [Dumoulin, Visen, 2018]



U-Net

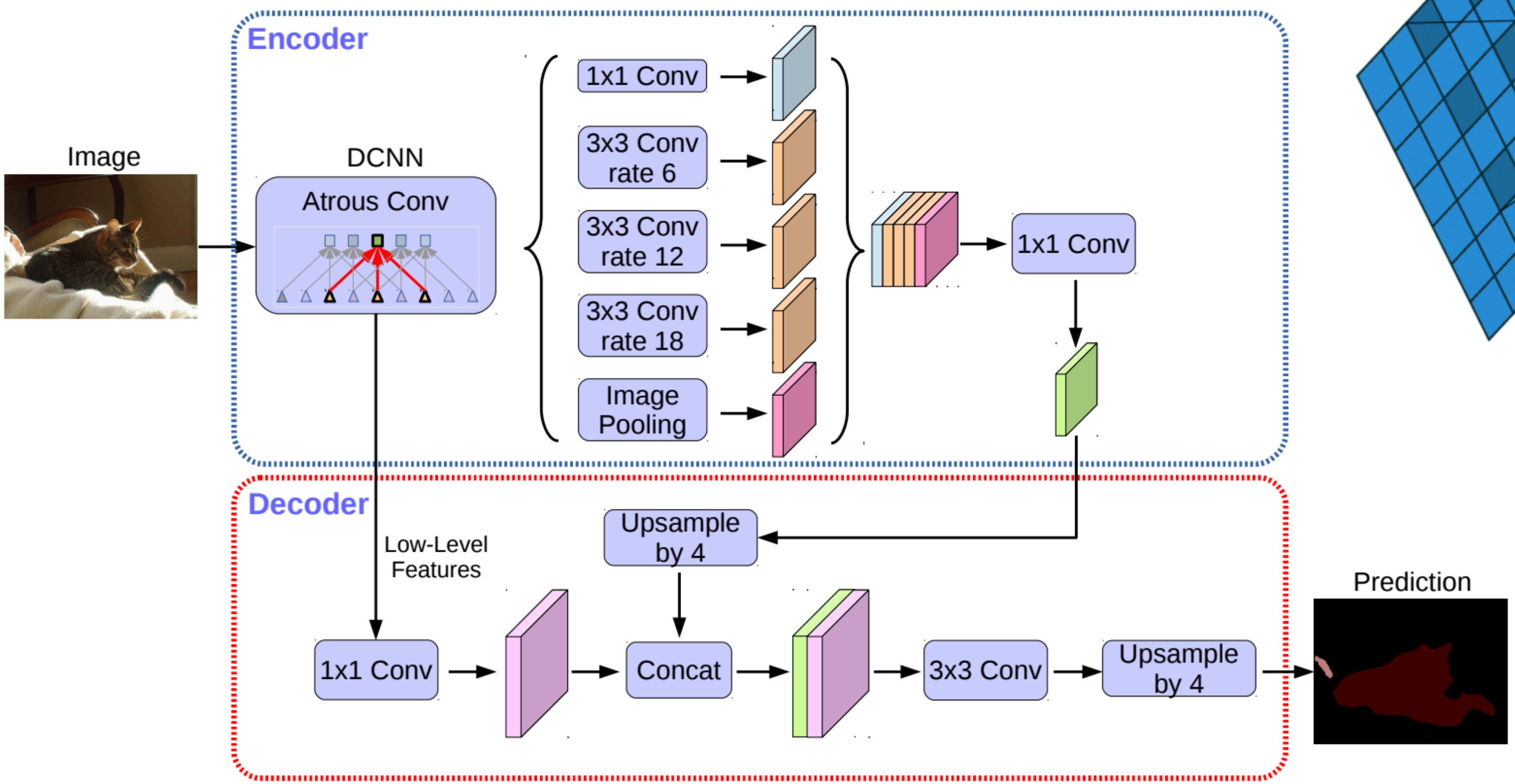
- Ronneberger, et al. U-Net: Convolutional Networks for Biomedical Image Segmentation, Medical Image Computing and Computer-Assisted Intervention, 2015

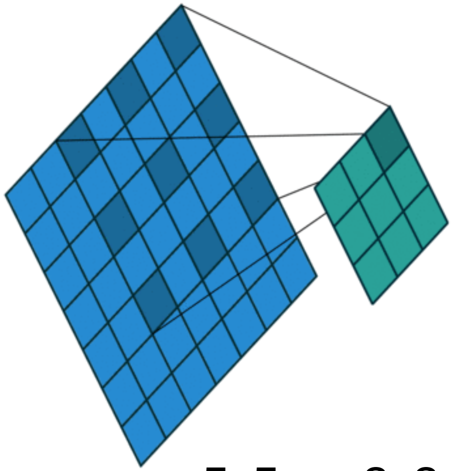


- Bahnik et al., Visually Assisted Anti-Lock Braking System. IEEE IV, 2020
  - Surface segmentation

DeepLab v3+

- Chen et al., Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, ECCV 2018.
- Atrous Convolutions (= with “holes”, dilated convolutions)
  - Same number of parameters with larger receptive field





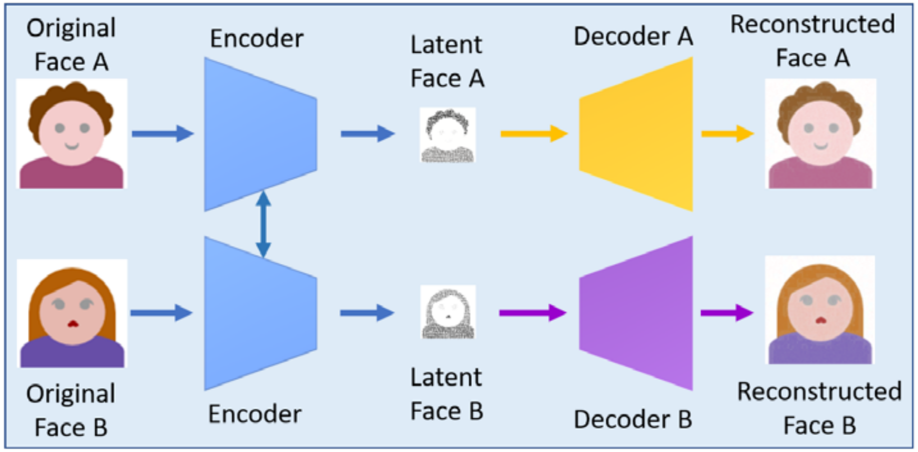
5x5 => 3x3 parameters

“Deeper” Insight into the Deep Nets

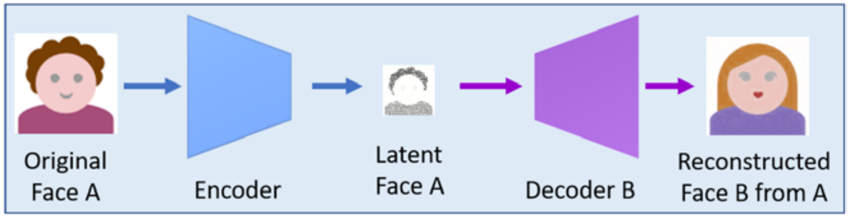
Deep Fake

- Seamless swapping a face in an image/video, e.g. [Nguyen et al., 2020]
- Auto-encoder architecture
  - Single shared encoder (to capture pose / expressions)
  - Two decoders (Source and Target to capture person's identity)

Training



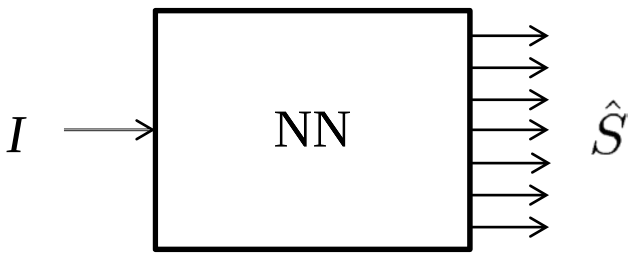
Deployment



- Controversy:
  - fake news, fake porn, …
- Deep fake detection

Deep Network Can Easily Be Fooled

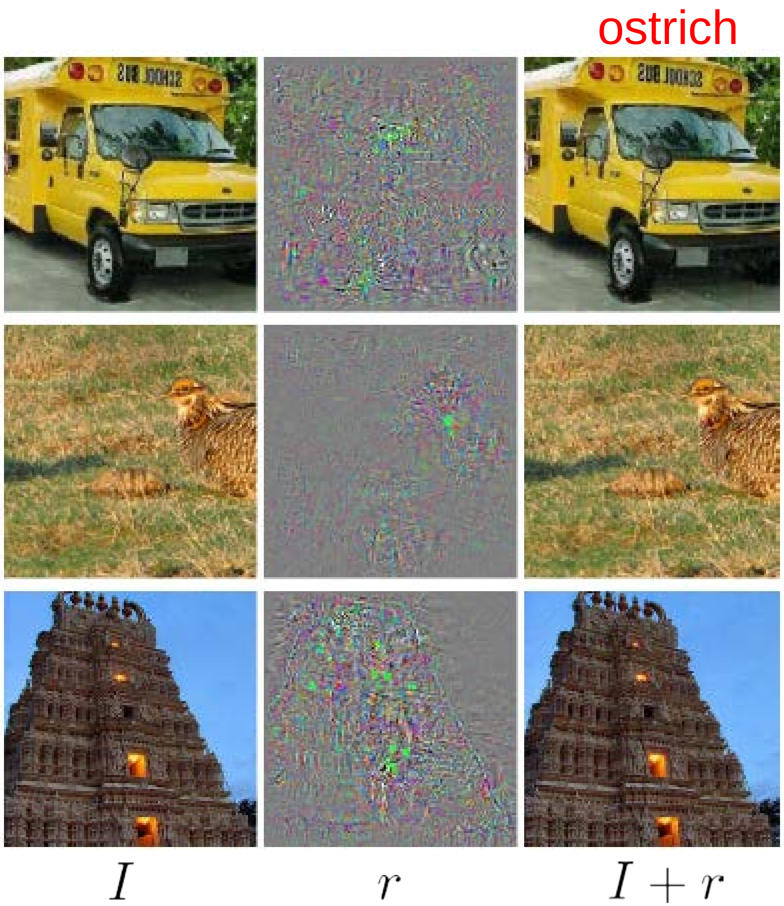
- Szegedy et al. Intriguing properties of neural networks. ICLR 2014
  - Small perturbation of the input image changes the output of the trained "well-performing" neural network
  - The perturbation is a non-random image, imperceptible for human



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

  - Optimum found by gradient descent

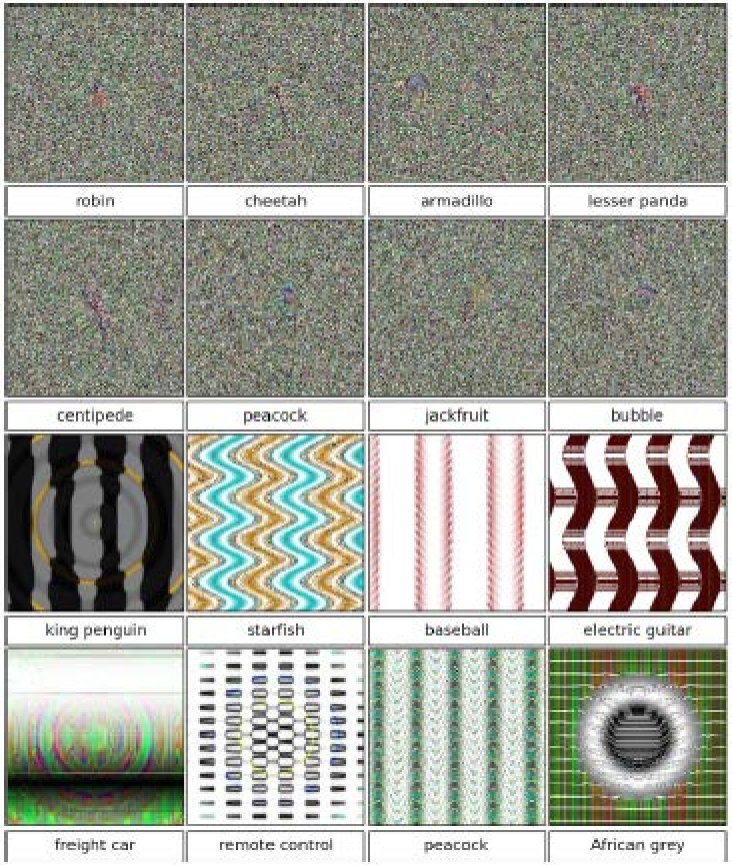
$$r^{t+1} = r^t - 2\gamma \left((\mathrm{NN}(I + r^t) - S) \frac{\partial \mathrm{NN}(I)}{\partial I} + \lambda r^t \right)$$



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

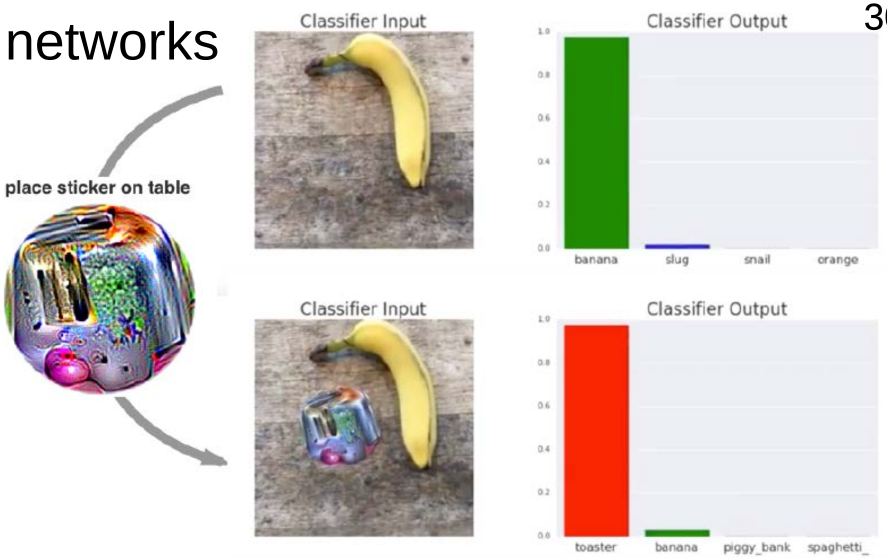
$$\min_{I} ||\mathrm{NN}(I) - S||^2$$



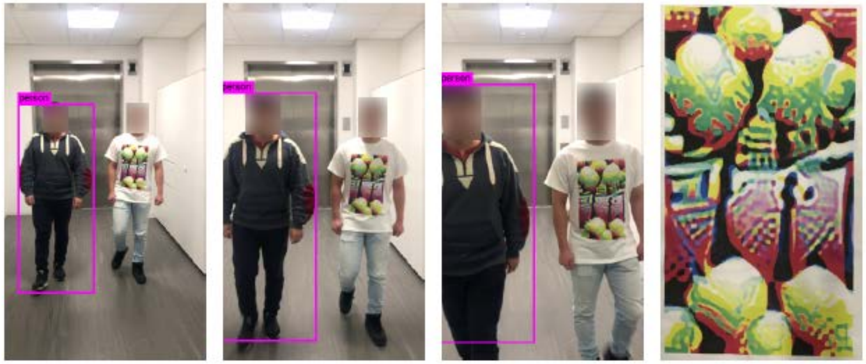
$\Rightarrow$ The images found do not have the natural image statistics

Deep Network Can Easily Be Fooled

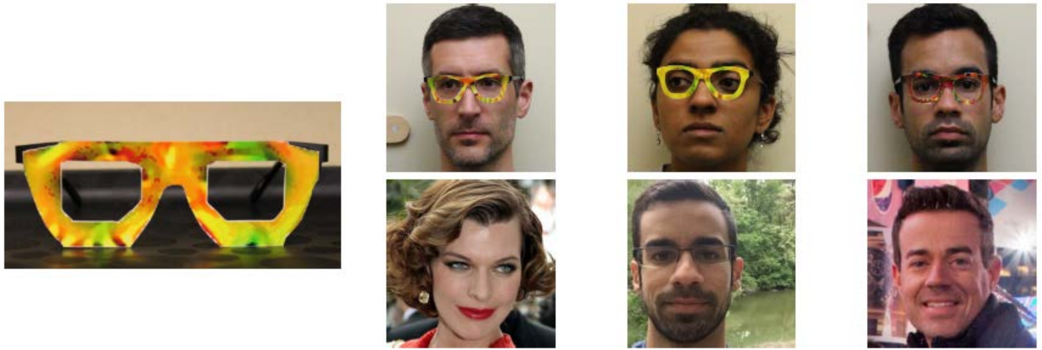
- Adversarial physical attacks on neural networks
  - Adversarial sticker [Brown-2018]



  - Adversarial T-shirt [Xu-2019]

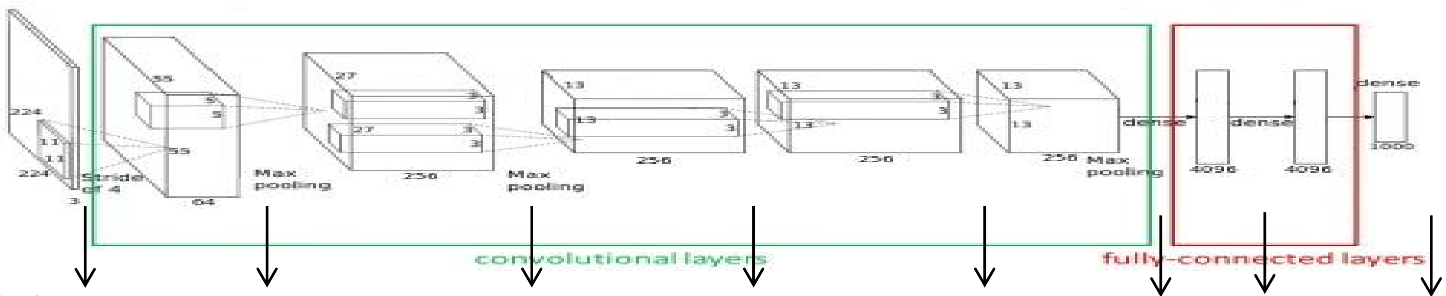


  - Adversarial glasses [Sharif-2016]



Visualization the Deep Nets

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



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

- Start from a random Image I
- Best match between features + image regularization (natural image prior)

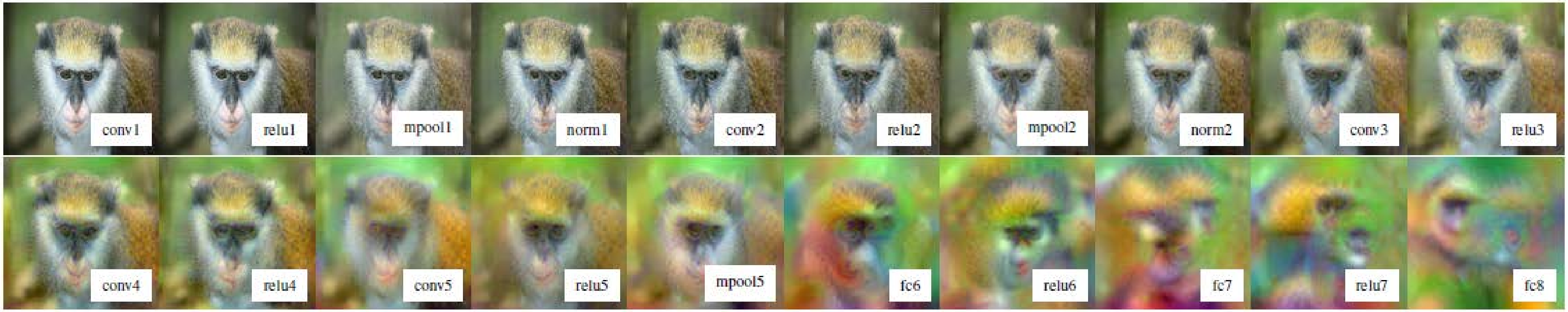
$$\min_I \{ ||\Phi_k(I) - \Phi_k^0||^2 + \lambda R(I) \}$$

- Total Variation regularizer (TV)

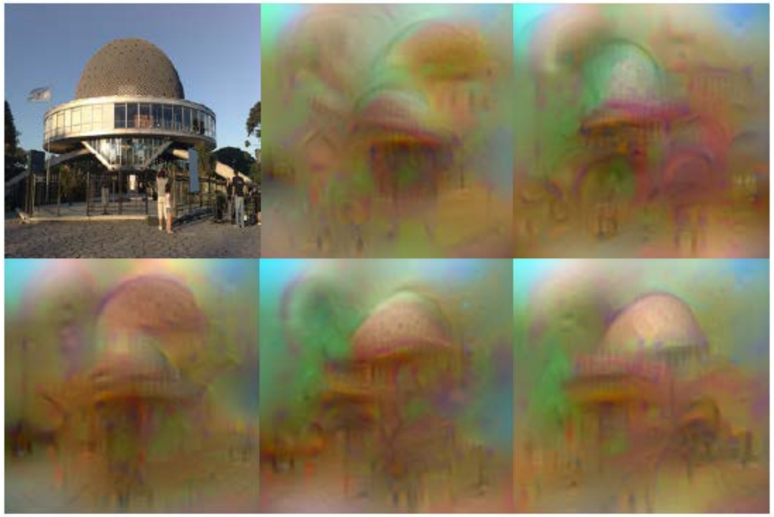
$$R(I) = \sum_{x,y} \Big(\big(\frac{\partial I(x,y)}{\partial x}\big)^2 + \big(\frac{\partial I(x,y)}{\partial y}\big)^2 \Big)^{\frac{\beta}{2}}$$

Visualizing the Deep Nets

- CNN reconstruction



  – Gradient descent from random initialization
  – Reconstruction is not unique



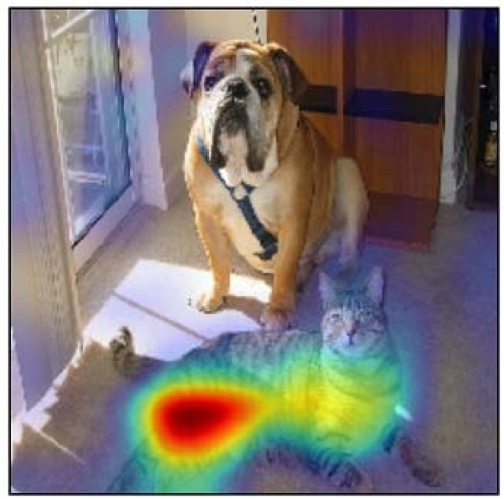
  ⇒ All these images are identical for the CNN

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

Verification what the deep net learned

- Deep nets often criticized for a lack of interpretability
- Grad-CAM: Visual Explanations from Deep Networks [Selvaraju-ICCV-2017]
  - GRADient weight Class Activation Mapping
  - Trianed model => Coarse localization map highlighting important regions for a class c

VGG “c=cat”



VGG “c=dog”

$$\alpha_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial NN(I)^c}{\partial \Phi_{ij}^k}$$

$$L^c_{\text{Grad-CAM}} = ReLU\left(\sum_k \alpha_k^c \Phi^k\right)$$

$\Phi_{i,j}^k$ …Feature tensor (last convolution layer)

i, j - spans spatial dimensions

k - spans channels

Deep Dream

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

$$\max_{I}\Big(||\mathrm{NN}(I)||^2 - R(I)\Big)$$

Credit: Eric Wayne

[video]

http://youtu.be/EjiyYtQIEpA

Deep Dream

- Maybe…

Salvador Dalí

Soft Construction with Boiled Beans (1936)

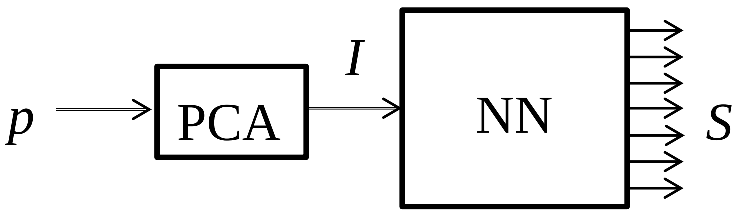
Swans Reflecting Elephants (1937)

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

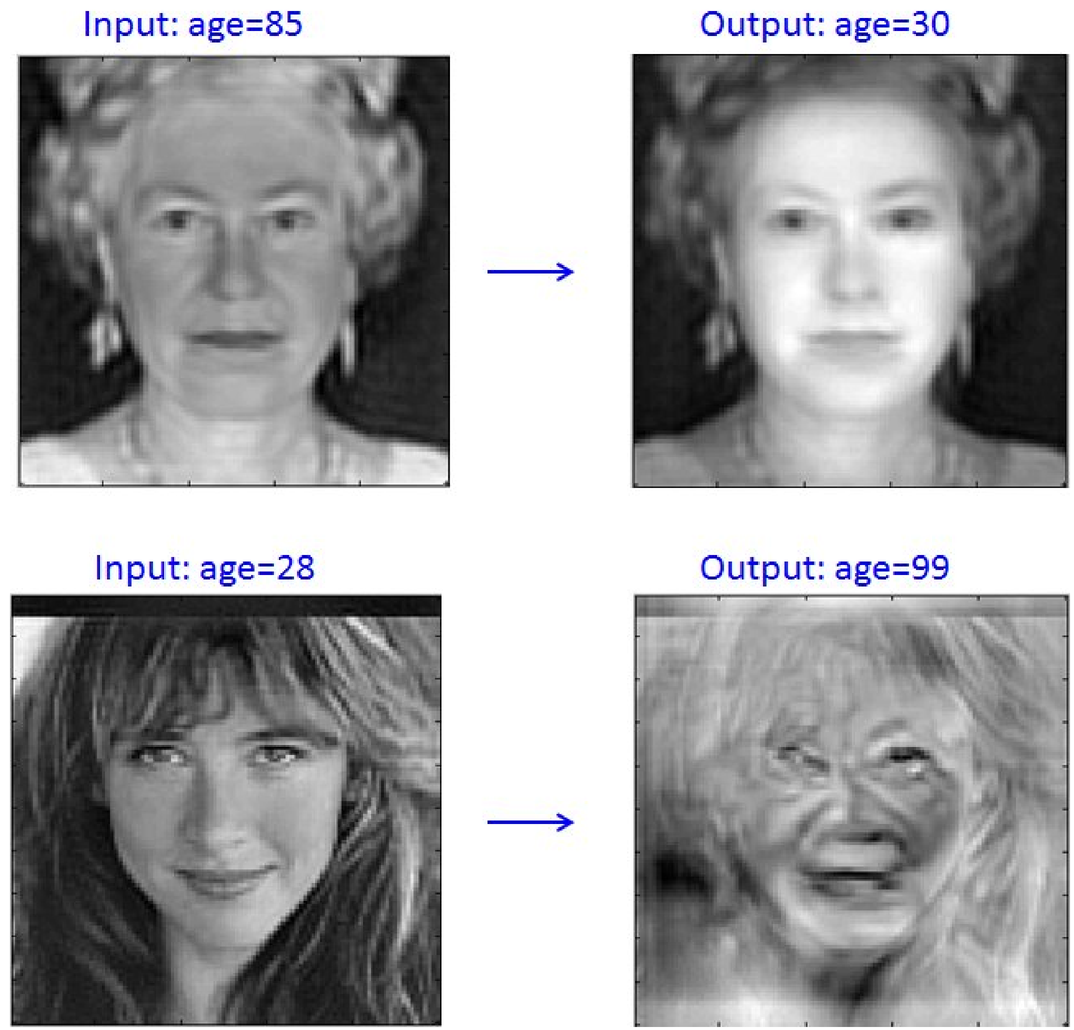
Hieronymus Bosch, Garden of Earthly Delights (~1510), [part]

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, 2015]

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

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

=> Understanding of the trained model

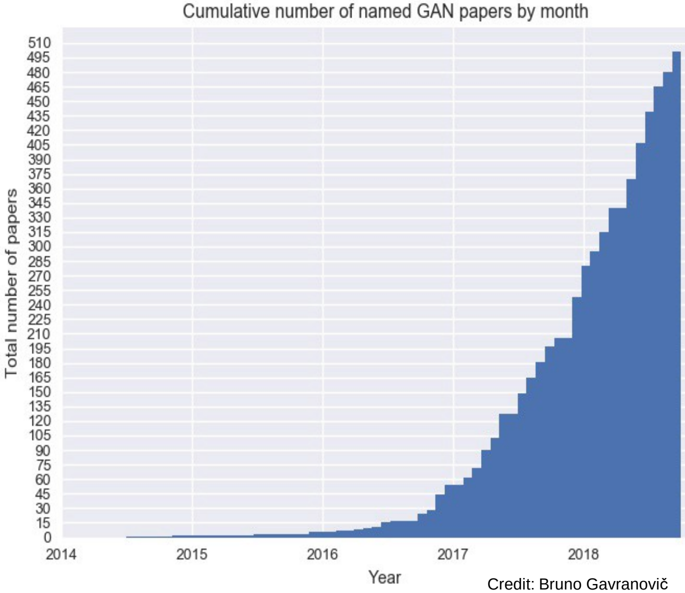
Generative Models

Generative Models

- Generate samples from a given complicated distribution (e.g. synthesis of photo-realistic images of various classes)

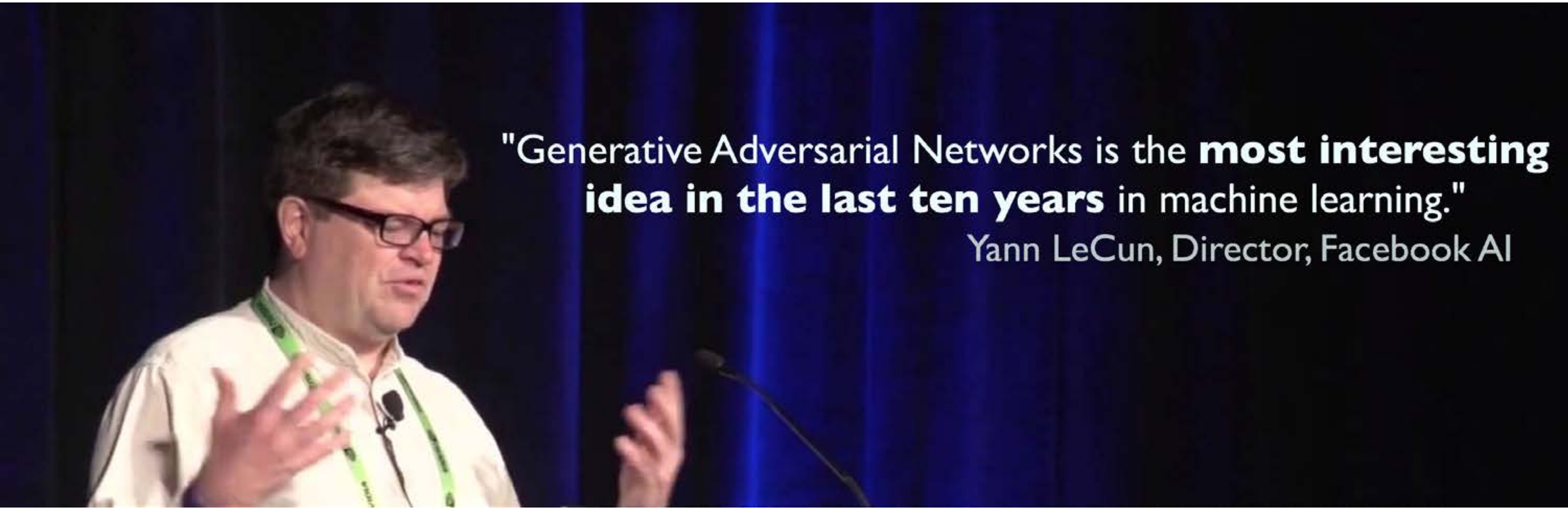
- Several approaches:
  1. Autoregressive models [Oord-2016]
  2. Variational Autoencoders [Kingma-2014]
  3. **Generative Adversarial Networks (GANs)** [Goodfellow-2014]

- Explosive interest in GANs
  – GAN Zoo



Credit: Bruno Gavranovič

Generative Adversarial Networks (GANs)



Generative Adversarial Networks (GANs)

[Goodfellow-2014]

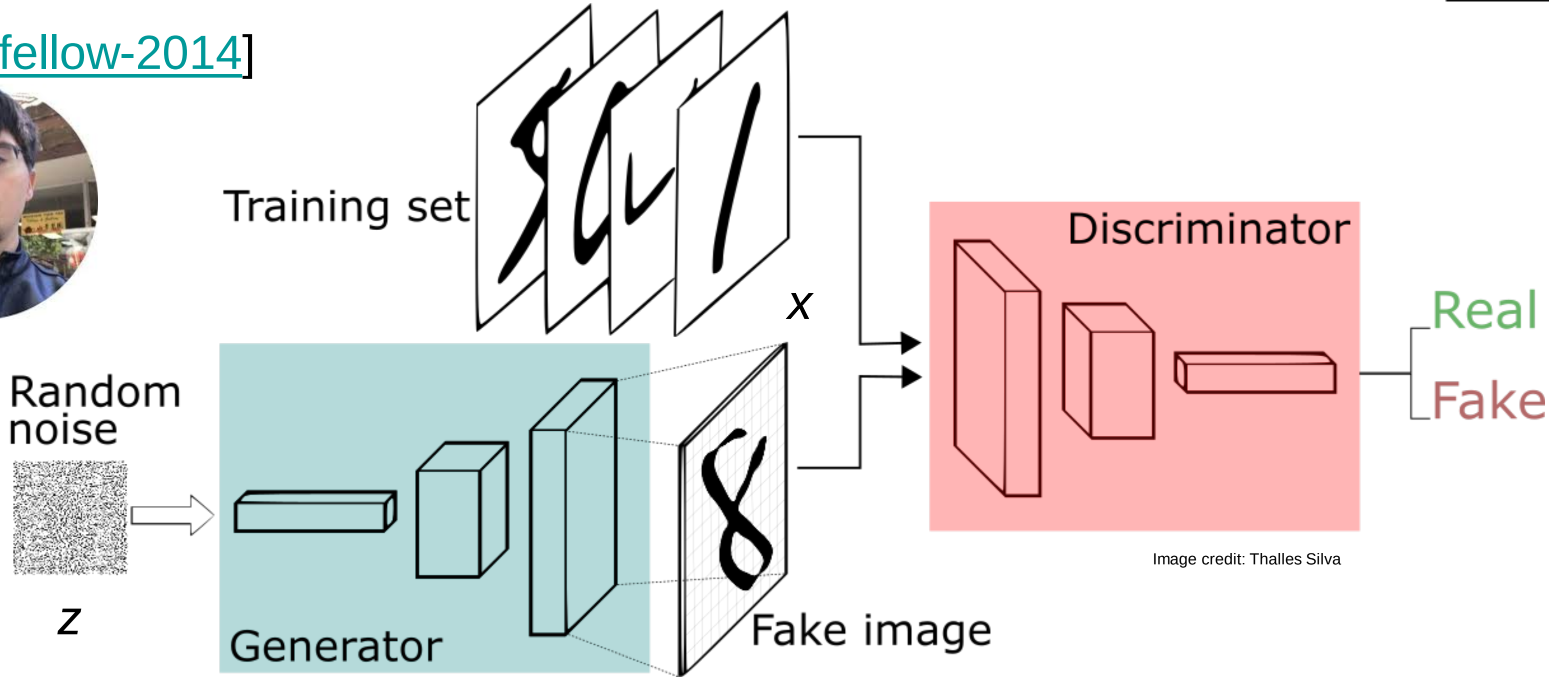
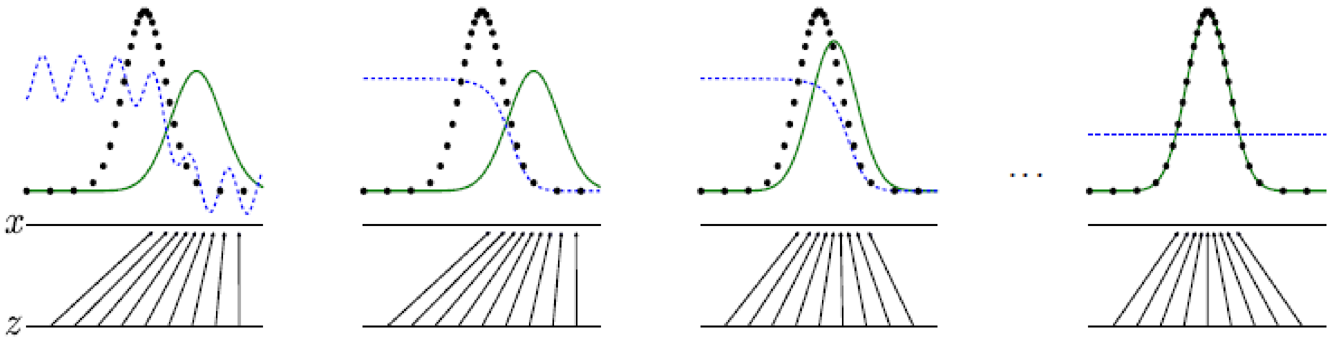


Image credit: Thalles Silva

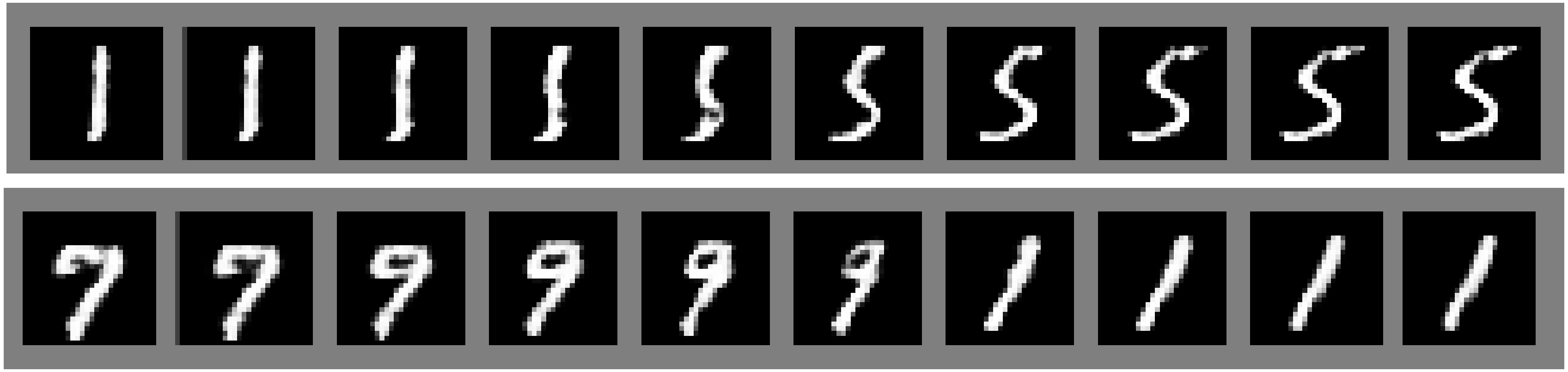
- Two networks: Generator G: $N(0,1)^k \rightarrow X$, Discriminator D: $X \rightarrow [0,1]$
- Min max game between G and D when training
  - The discriminator tries to distinguish generated and real samples
  - The generator tries to fool the discriminator

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Generative Adversarial Networks (GANs)



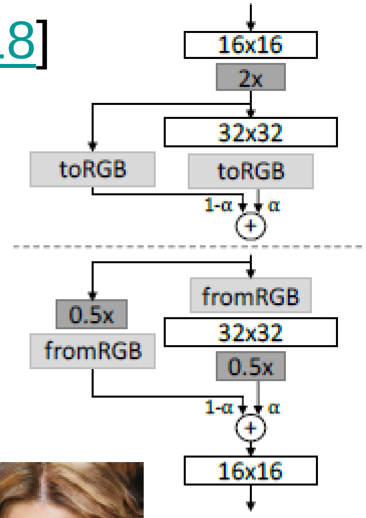
- Seems to capture the image manifold
  - Smooth transitions when interpolating in the latent space

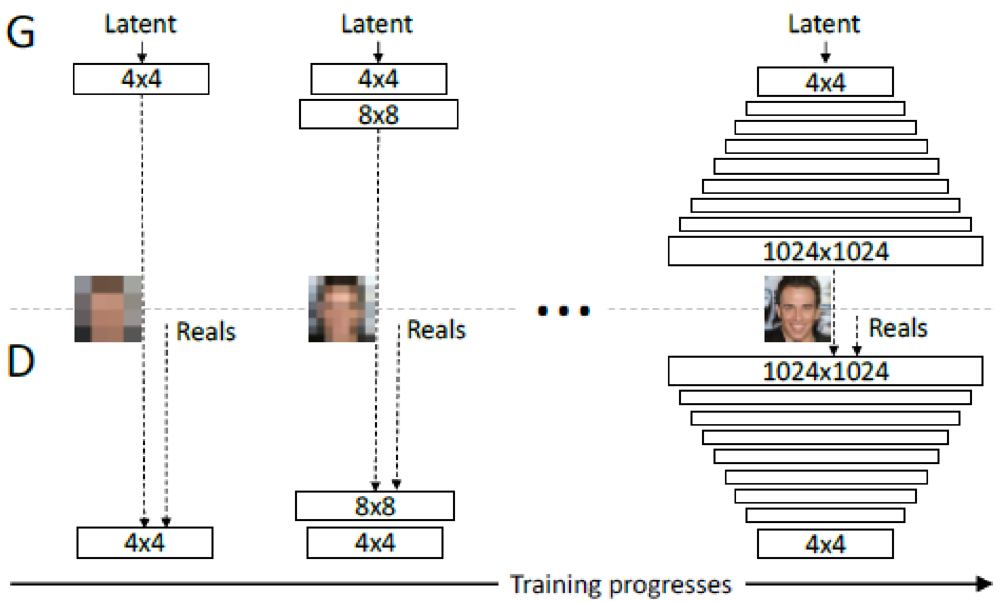


- However:
  - The training is fragile (alternating optimization), mode collapse
  - Did not work well for high-resolution (until recently)

High resolution GANs

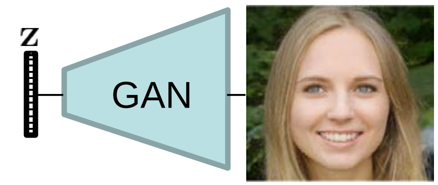
- Synthesis of 1024x1024 face images [Nvidia-ProGAN-2018]
- Trained from CelebA-HQ dataset 30k images
- Progressive training
  - Complete GAN for low-resolution (4x4)
  - Upsample, concatenate with res-net connections
  - Train everything end-to-end



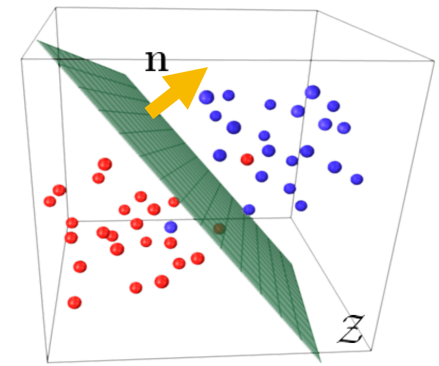


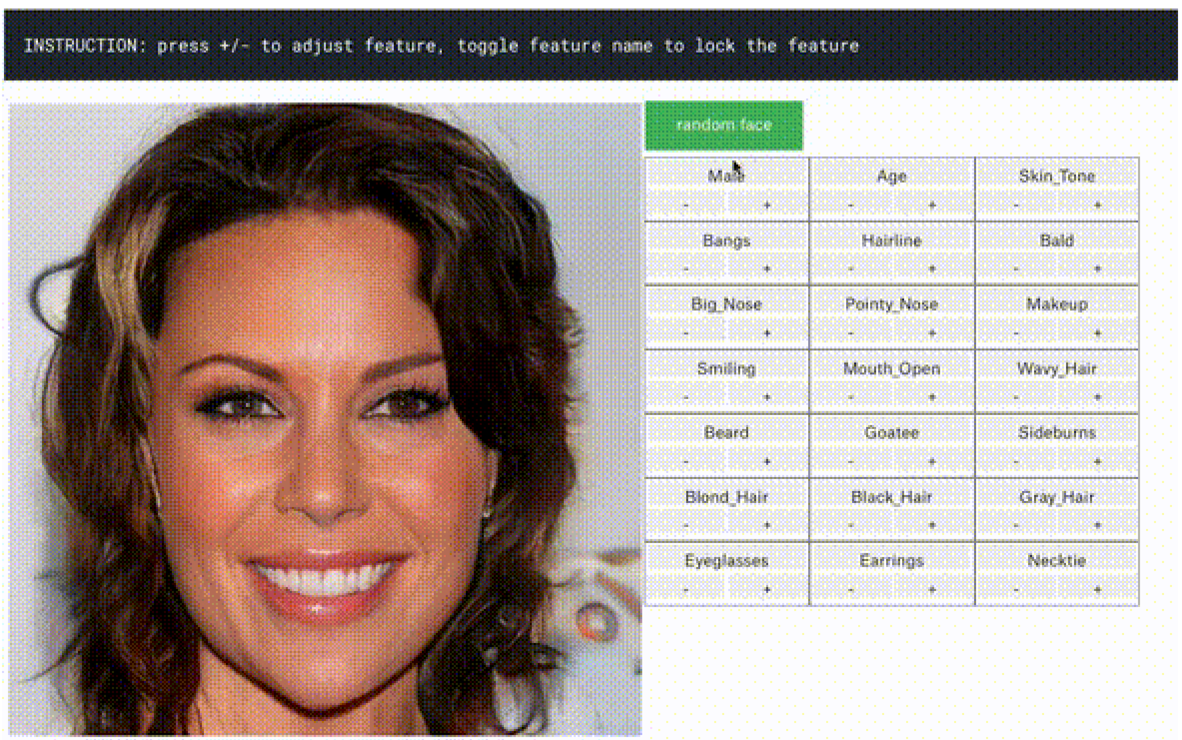
- Follow-up paper [Nvidia-StyleGAN-2019, Nvidia-2020, Nvidia-2021]
  - Multi-layer style transfer, training from 70k Flicker dataset, “hyper-realistic”

GAN – latent space manipulation



- Every $\mathbf{z}$ from input distribution gives a realistic image
- Finding semantic direction in the latent vector space
  - Train a linear binary classifier on labeled set $(\mathbf{z}_i, y_i)$
  - Normal of the discriminative hyperplane is the semantic direction
- Semantic Editing / “Manipulation” $\mathbf{z} = \mathbf{z}_0 + \alpha\mathbf{n}$



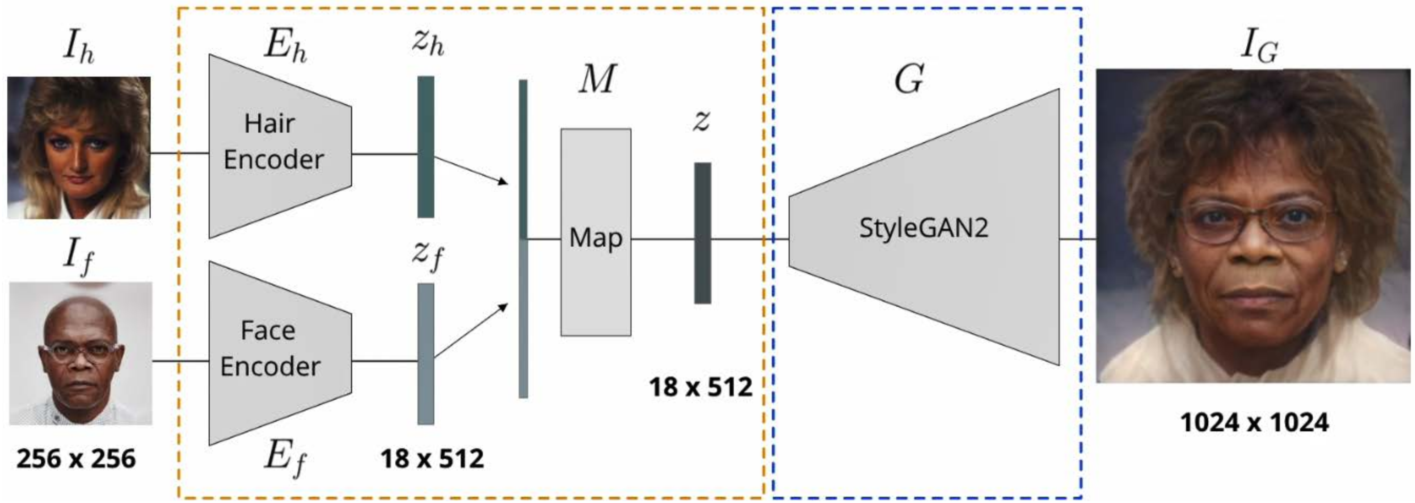


[demo]

[Abdal-SIGGRAF-2021]

Hairstyle Transfer using StyleGAN

- Fully automatic hairstyle transfer, unaligned portraits [Šubrtová-FG-2021]



[video1]
[video2]

- Basic idea: Train two encoders (Hair, face) + fixed StyleGAN decoder
- Editing in hairstyle latent space

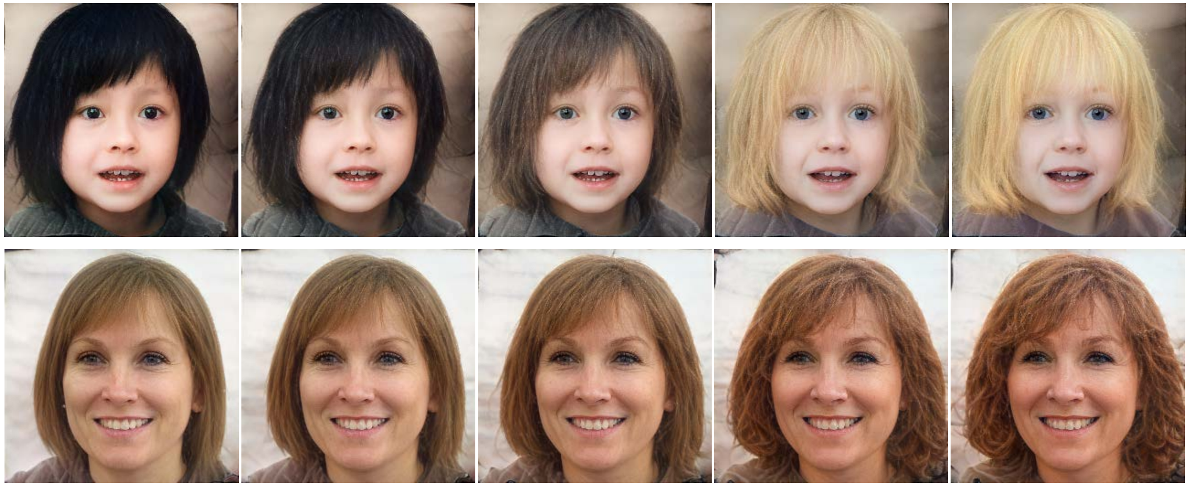
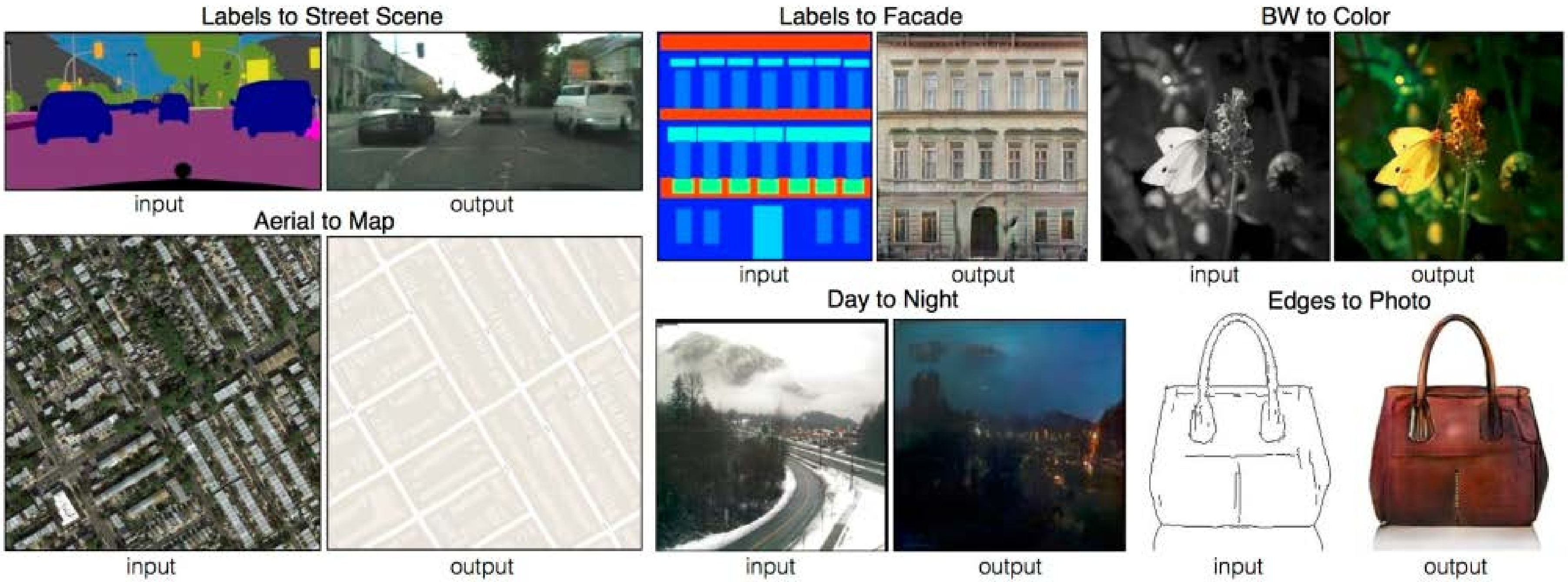


Image to Image Translation

- Transfer image between domains [Isola-Zhu-Zhou-Efros-2017]

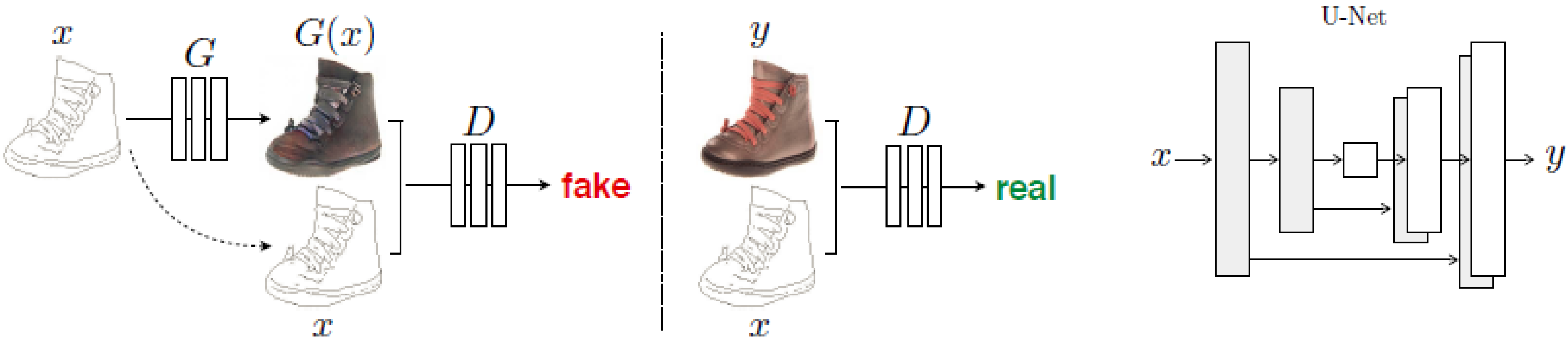


- Many applications [pix2pix], Super-resolution [Šubrtová-2018]

16x16 256x256 (predicted) 256x256 (ground-truth)

Image to Image Translation

- Combines fully convolutional net training with (conditional) GAN



$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

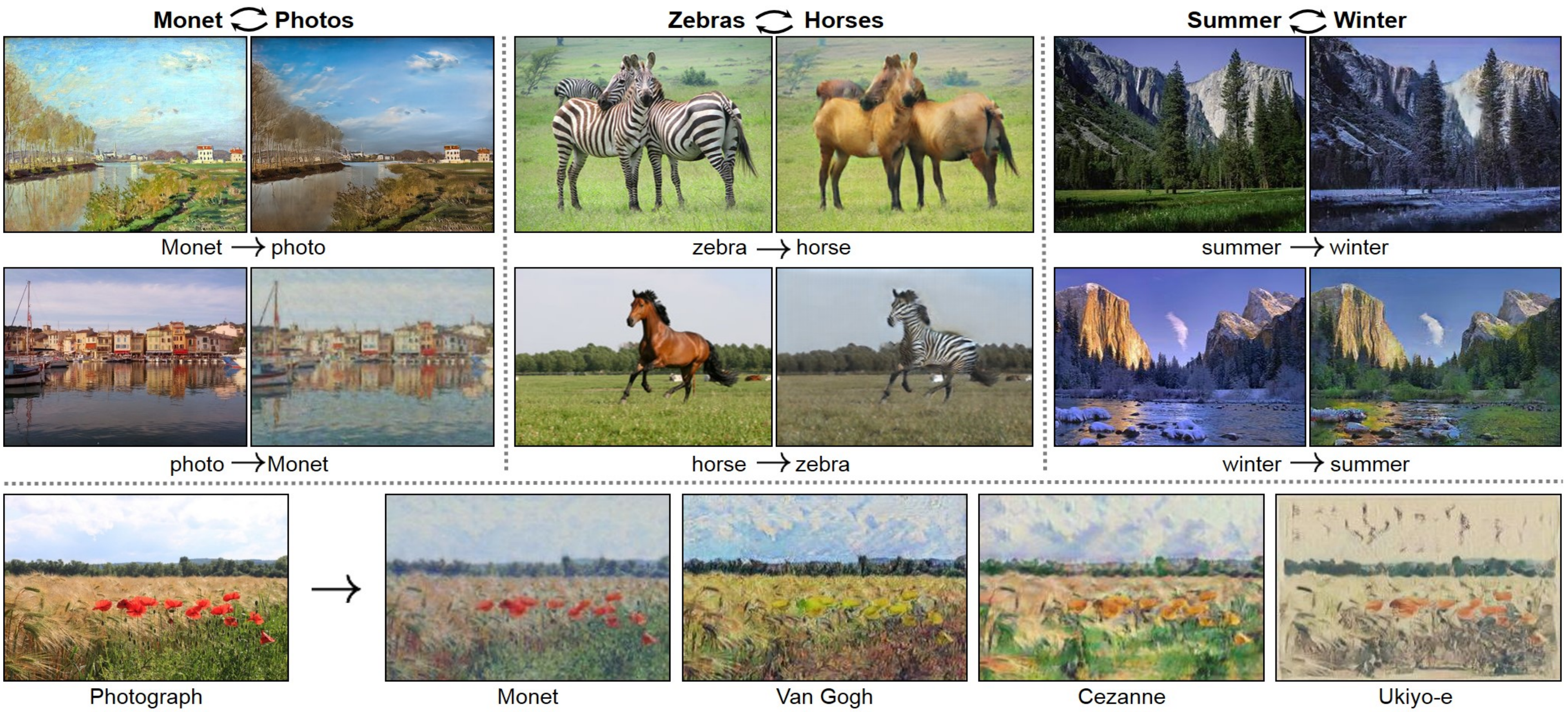
$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1]$$

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z)))]$$

- Difficulties with imposing variability (only via dropout when testing)
- Training needs pixel-to-pixel source and target image correspondences

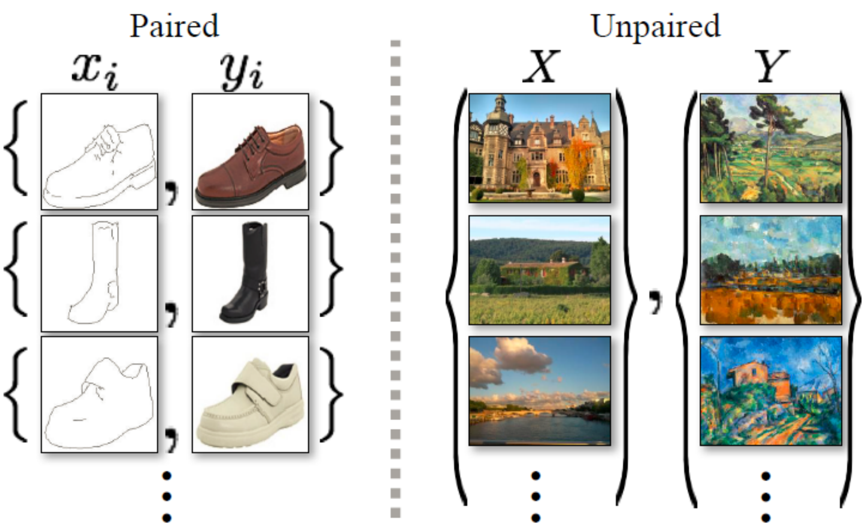
Cycle GAN

- Translating without pix-to-pix correspondences [Zhu-Park-Isola-Efros-2017]



Cycle GAN

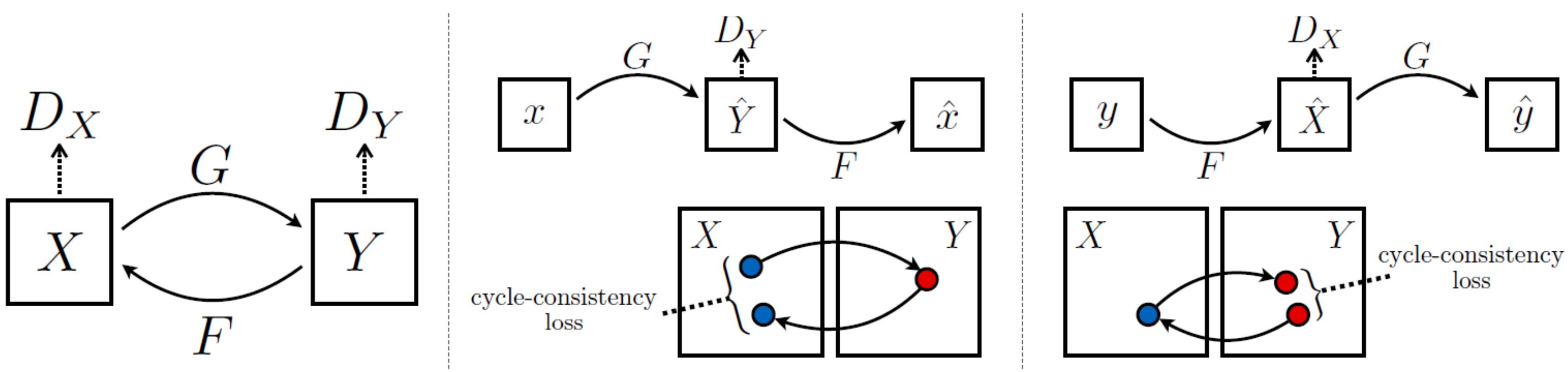
- Unpaired set of images to train the translation



$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F),$$

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$$

- Cycle consistency



What was not mentioned…

What was not mentioned…

- **Recurrent NNs**
  - Processing sequences, prediction, image captioning, etc.
  - Fei-Fei Li, Andrej Karpathy @ Stanford
  - **(Now seem outperformed by Transformers)**

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."

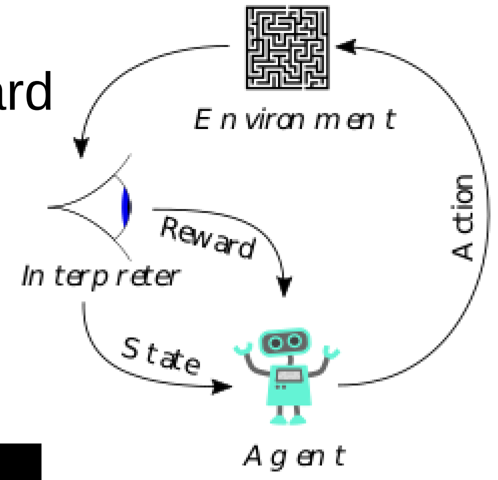
Linux source code

```
/*
 * 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."

What was not mentioned…

- **Reinforcement Learning**
  - Agent interacts with environment to maximize reward
  - Learning to play Atari games
  - Learning to drive
  - Learning to walk, maneuvering, etc.
  - “hot-topic” in robotics



ATARI

Conclusions

- Fathers of the Deep Learning Revolution Receive Turing Award 2018:

- 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?