Deep Learning Lab: Computer Vision

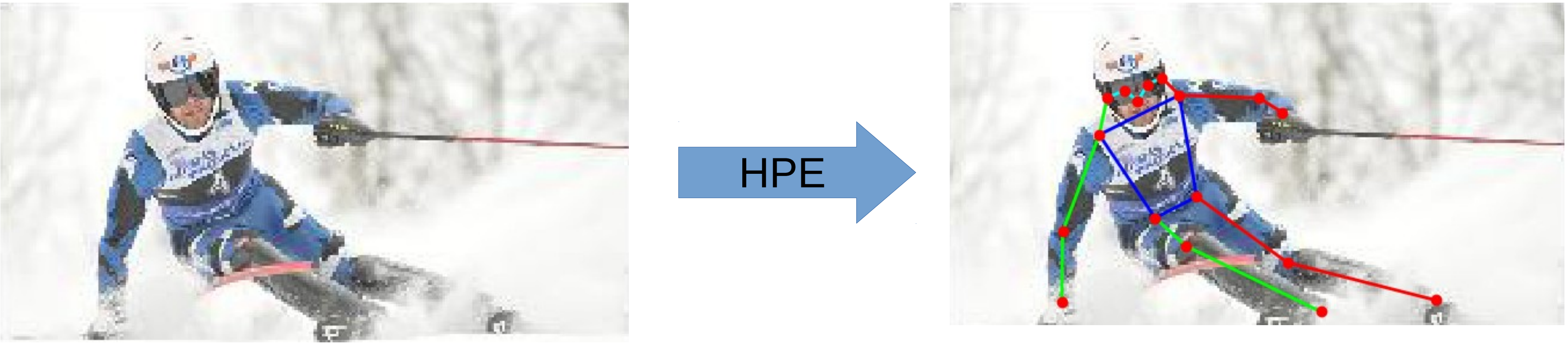
Christian Zimmermann and Silvio Galesso

Outline

- Introduction to Human Pose Estimation (HPE)
- Single HPE
- Multi HPE
  - Top down
  - Bottom up
- Semantic Segmentation
- Exercise

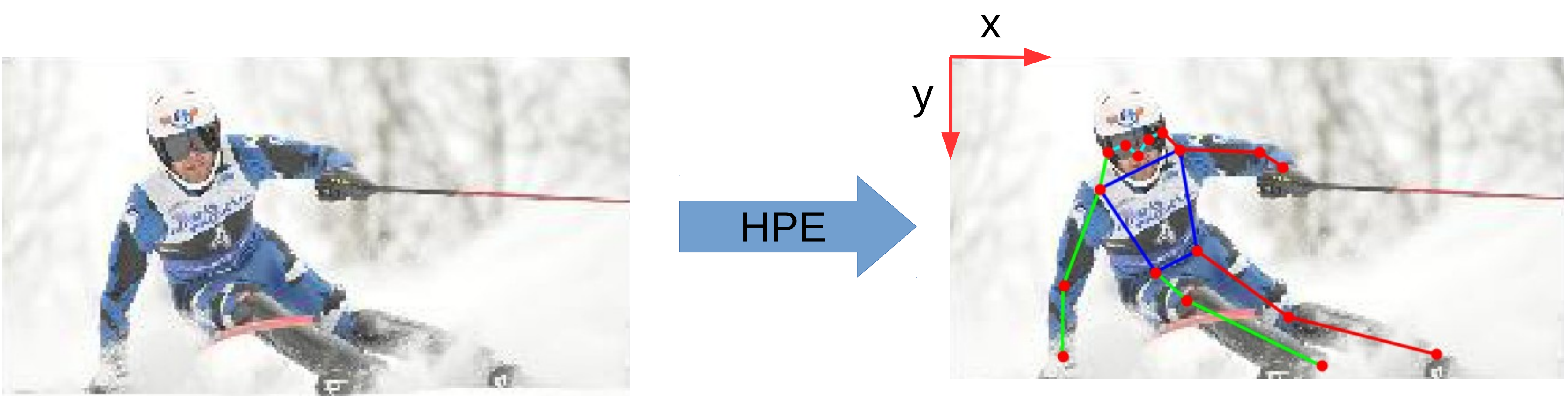
Intro: What is HPE?

- Given a single color image infer body pose:



Intro: What is HPE?

- Given a single color image infer body pose:

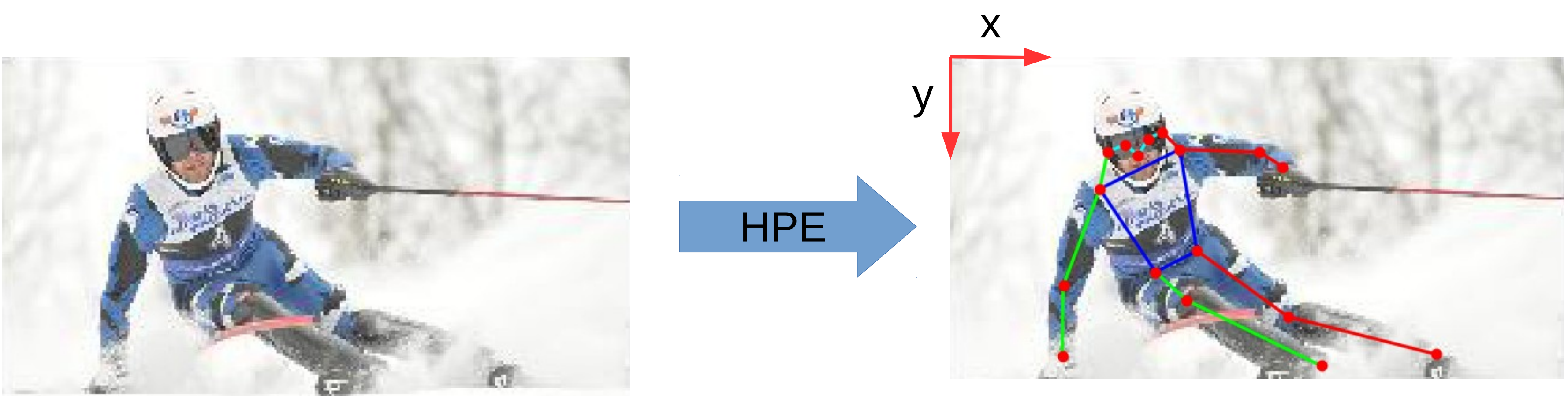


- 2D pose *P* is defined as:

$$P = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} = \begin{pmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix} \in \mathbb{R}^{n x 2}$$

Intro: What is HPE?

- Given a single color image infer body pose:



- 2D pose P is defined as:

$$P=\begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix}=\begin{pmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{pmatrix}\in\mathbb{R}^{n x 2}$$

f.e. “nose”

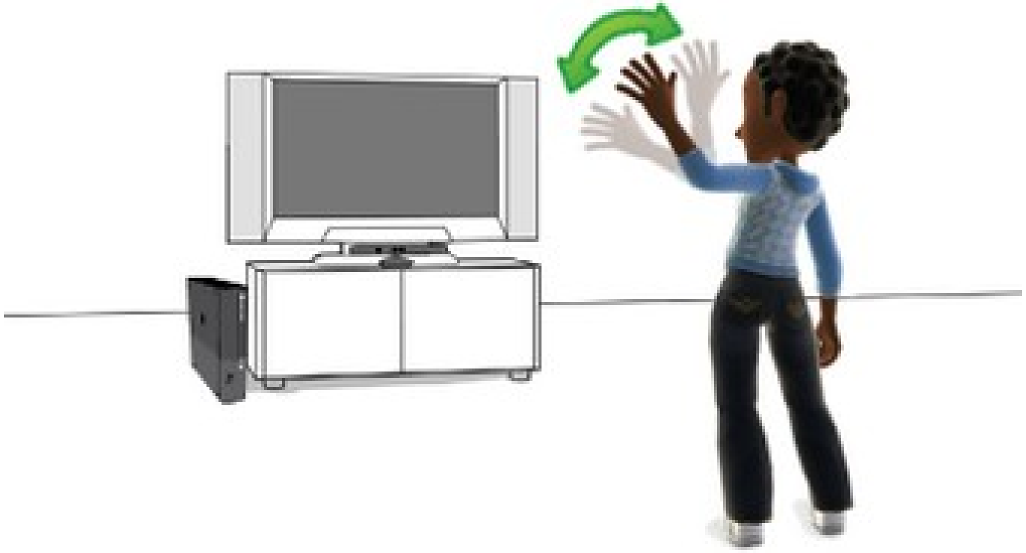
Intro: Why HPE?

- Human machine interaction:
  - Autonomous driving: Infer People and their heading direction and intentions

From: Kreiss et al., PifPaf: Composite Fields for Human Pose Estimation, CVPR 2019

Intro: Why HPE?

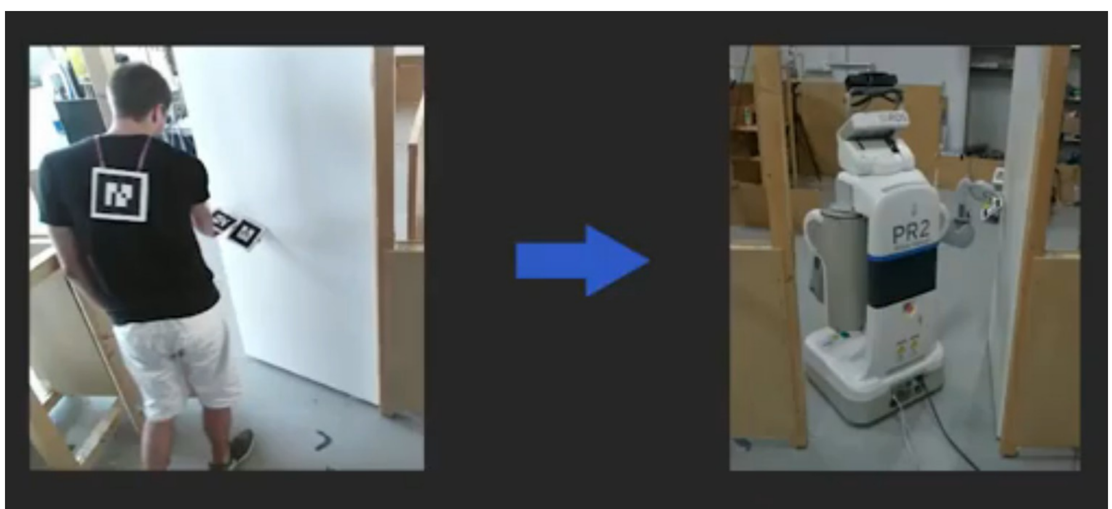
- Human machine interaction:
  - Autonomous driving: Infer People and their heading direction and intentions
  - Pose based gaming: Microsoft Xbox Kinect



From: xbox.com

Intro: Why HPE?

- Human machine interaction:
  - Autonomous driving: Infer People and their heading direction and intentions
  - Pose based gaming: Microsoft Xbox Kinect
  - In robotics: Learning from demonstration



From: Zimmermann et al., 3D Human Pose Estimation in RGBD Images for Robotic Task Learning, ICRA 2018

Intro: Why HPE?

- Human machine interaction
- Quantify movement:
  - Sport action analysis
    - Track players during sports
  - Medicine
    - Whats the stage of the ALS disease?

Intro: What makes it hard?

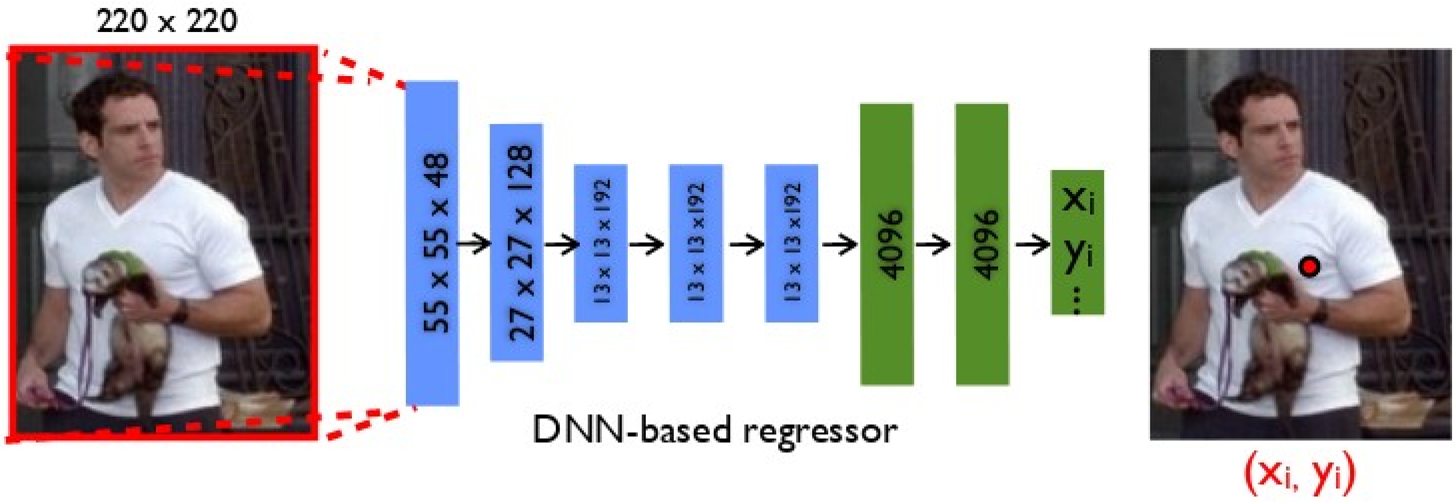
- Large variation in appearance
- Ambiguities

Intro: What makes it hard?

- Large variation in appearance
- Ambiguities
- Occlusions
- Crowding

Single HPE: Regression

- Directly regress Cartesian image coordinates
- Network outputs one vector of coordinates



From: Toshev et al., DeepPose: Human Pose Estimation via Deep Neural Networks, CVPR 2014

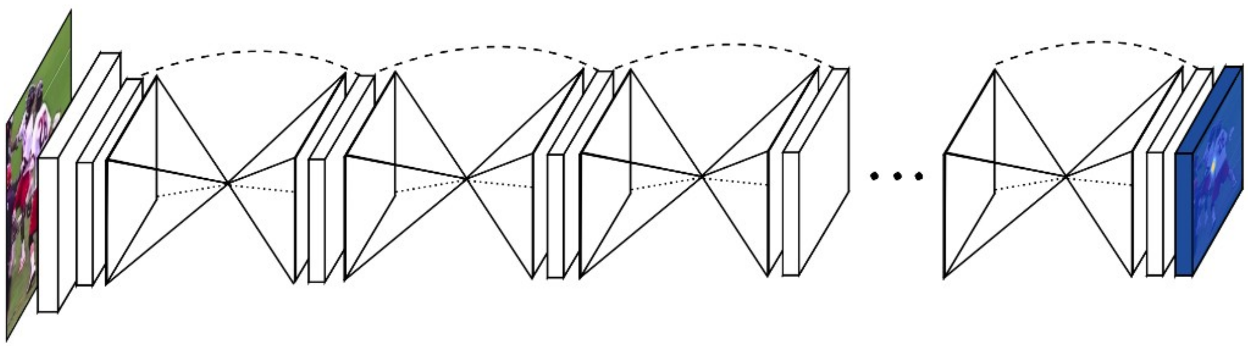
$$L = \sum_i \left(\| P_i - \hat{P}_i \|_2 \right)^2$$

Prediction
Ground truth

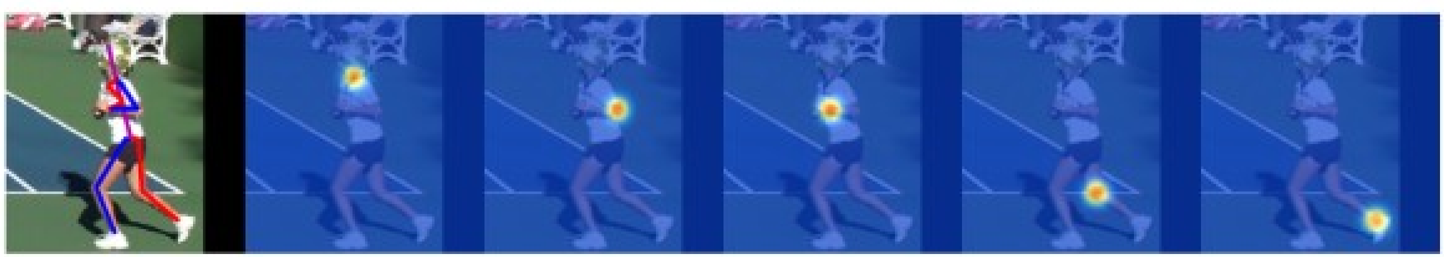
Single HPE: Scoremap

- Network estimates per pixel keypoint likelihood
- For each keypoint there is one map
- Ground truth maps are created from point annotations

$I \in \mathbb{R}^{NxNx3}$



$S \in \mathbb{R}^{MxMxK}$

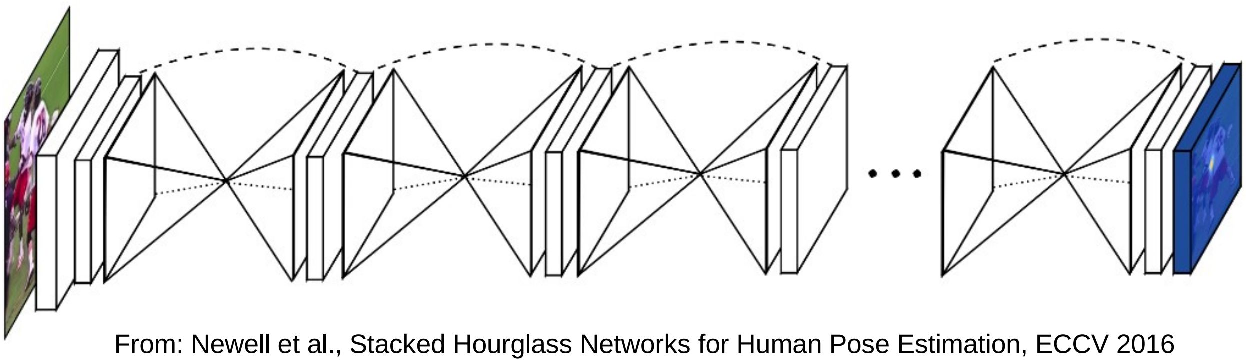


From: Newell et al., Stacked Hourglass Networks for Human Pose Estimation, ECCV 2016

Single HPE: Scoremap

- Network estimates per pixel keypoint likelihood
- For each keypoint there is one map
- Ground truth maps are created from point annotations

$I \in \mathbb{R}^{NxNx3}$



$S \in \mathbb{R}^{MxMxK}$

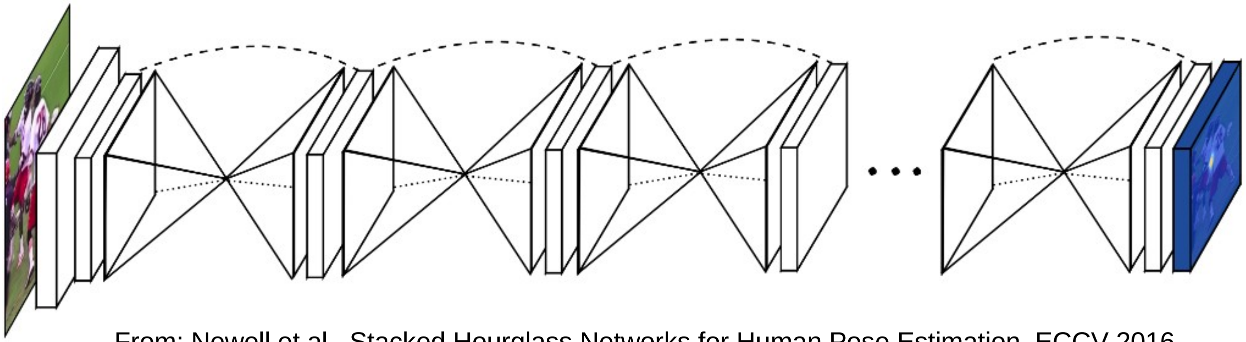
From: Newell et al., Stacked Hourglass Networks for Human Pose Estimation, ECCV 2016

$$L = \sum_{i}^{K} \left(\| \underbrace{S_i}_{\text{Prediction}} - \underbrace{\hat{S}_i}_{\text{Ground truth}} \|_2 \right)^2$$

Single HPE: Scoremap

- Network estimates per pixel keypoint likelihood
- For each keypoint there is one map
- Ground truth maps are created from point annotations

$I \in \mathbb{R}^{NxNx3}$



$S \in \mathbb{R}^{MxMxK}$

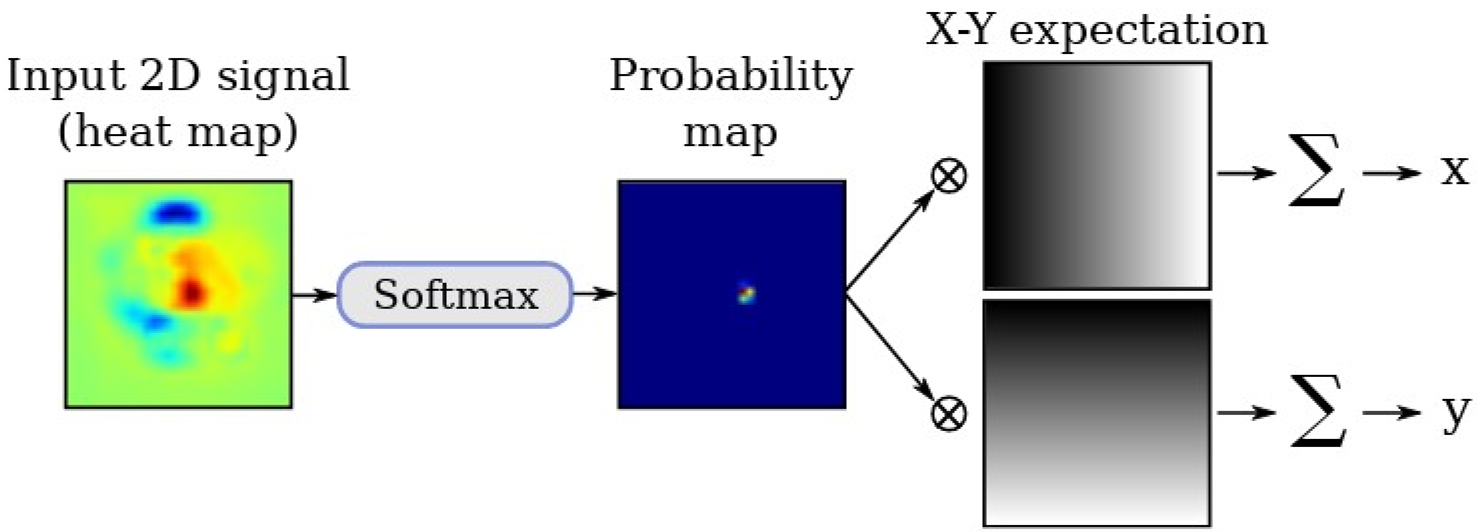
From: Newell et al., Stacked Hourglass Networks for Human Pose Estimation, ECCV 2016

$$L = \sum_{i}^{K} (\|S_i - \hat{S}_i\|_2)^2$$

$$\hat{S}_i(\tilde{p}) = \exp\left(\frac{-\|\tilde{p} - \hat{p}\|_2}{\sigma^2}\right)$$

Single HPE: Softargmax

- Heat maps are learned implicitly
- Softmax squashed map into a probability distribution
- Elementwise multiplication and summation reduces to predicted coordinate



From: Luvizon, 2d/3d pose estimation and action recognition using multitask deep learning, CVPR 2018

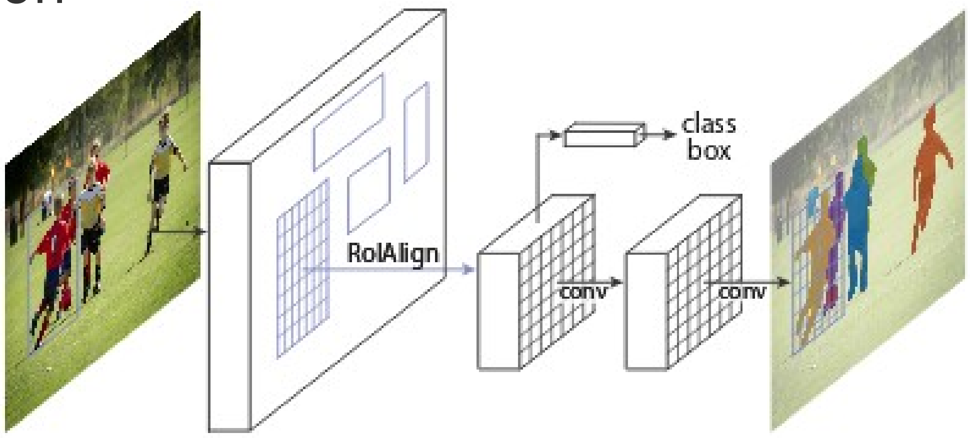
Multi HPE

Multi HPE

- Top-Down
  - Detects persons first
  - Estimates pose for each person independently
  - Suffers early commitment
  - Runtime scales linear in #people
  - Struggles when people crowd
- Bottom-Up
  - Detects keypoints first
  - Subsequently groups keypoints into indivuals
  - Makes keypoint detection harder because of less prior knowledge
  - Extensive grouping is NP hard problem

Top-down approach

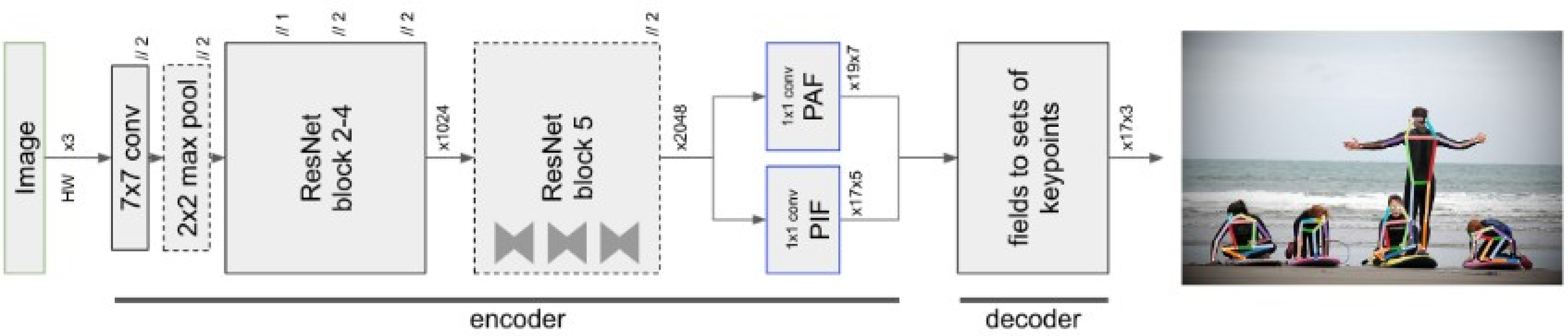
- Example for top-down: Mask R-CNN
  - First part of the network detects bounding boxes
  - Then pool features from each bounding box and apply a sub-network (‘head’) on them
  - There are heads for classification, segmentation and pose estimation



From: He et al., Mask R-CNN, ICCV 2017

Bottom-up approach

- Single network that estimates two entities:
  - Keypoint locations (**P**art **I**ntensity **F**ield, also called Scoremap) → Gives joint estimates
  - Association scores between keypoints that should form a limb (**P**art **A**ffinity **F**ields) → Enables grouping



From: Kreiss et al., PifPaf: Composite Fields for Human Pose Estimation, CVPR 2019

Bottom-up approach

- Part Intensity Fields
  - Likelihood at each location if the keypoint is present
  - One Scoremap per keypoint needed

From: Kreiss et al., PifPaf: Composite Fields for Human Pose Estimation, CVPR 2019

Bottom-up approach

- Part Affinity Fields
  - Vector field pointing in the direction from ‘start’ to ‘end’ of a limb
  - Two maps per keypoint (one for each vector component)

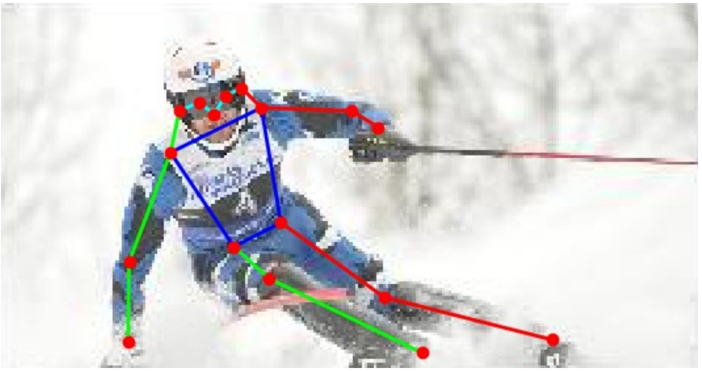
From: Cao et al., Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, CVPR 2017

Bottom-up approach

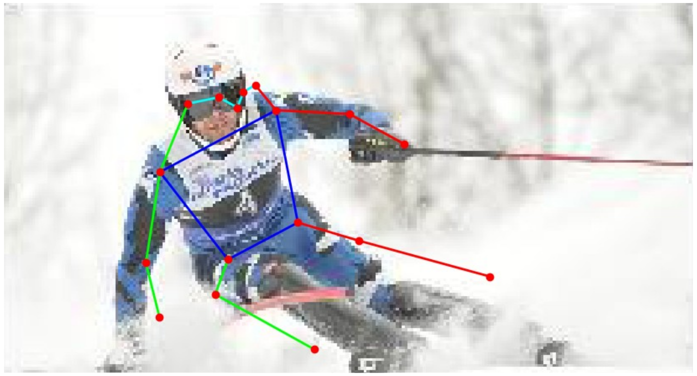
- Grouping keypoint candidates to instances
  - Finding the optimal parse of from the detected keypoint candidates is NP-Hard.
  - Therefore relax complete matching to greedy bipartite graph matching: i.e. match one limb at a time
- Practical implementation:
  - Start with the most confident keypoint locations
  - Greedy growing of the person instance using the PAF based score

Evaluation

- Common measure: Mean per joint position error (MPJPE)



Ground truth



Prediction

$$MPJPE = \frac{1}{|V|} \sum_{i \in V}^{K} \| p_i - \hat{p}_i \|_2$$

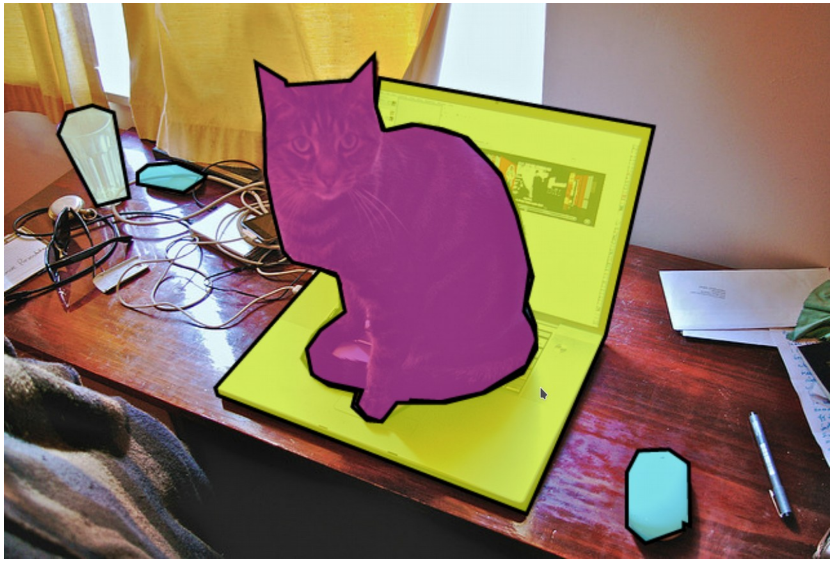
Set of visible keypoints

Semantic Segmentation

Image Segmentation

Definition: partitioning the image into coherent regions/subsets of pixels

Input image

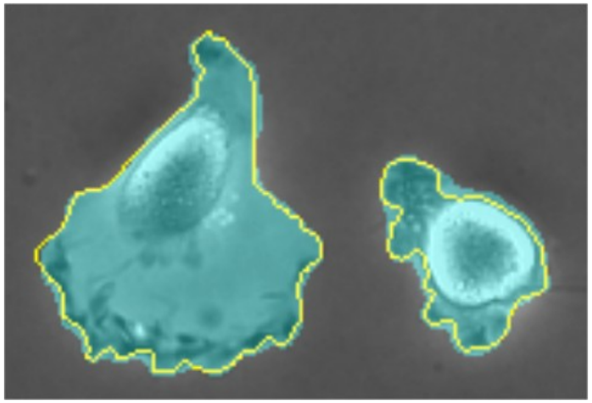


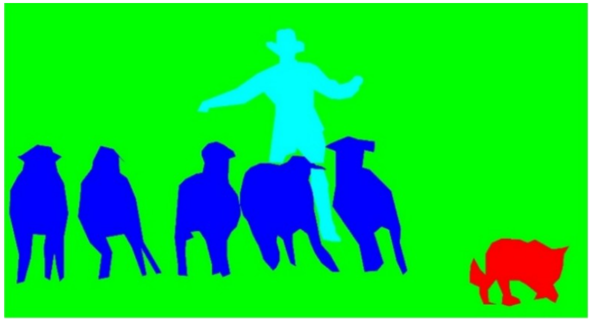
Segmentation mask

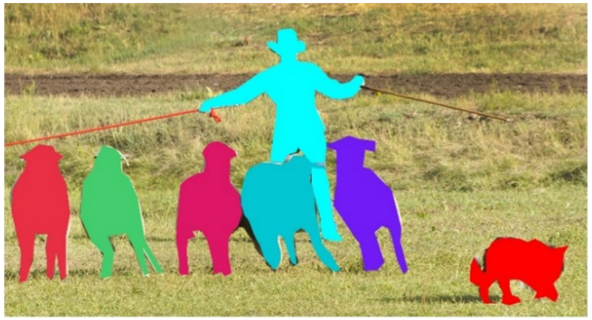
Source: COCO dataset, http://cocodataset.org/#explore

Segmentation Tasks

- Binary segmentation
- Semantic segmentation
- Instance segmentation







- Assign a class to each pixel
- 2 classes: foreground/ background

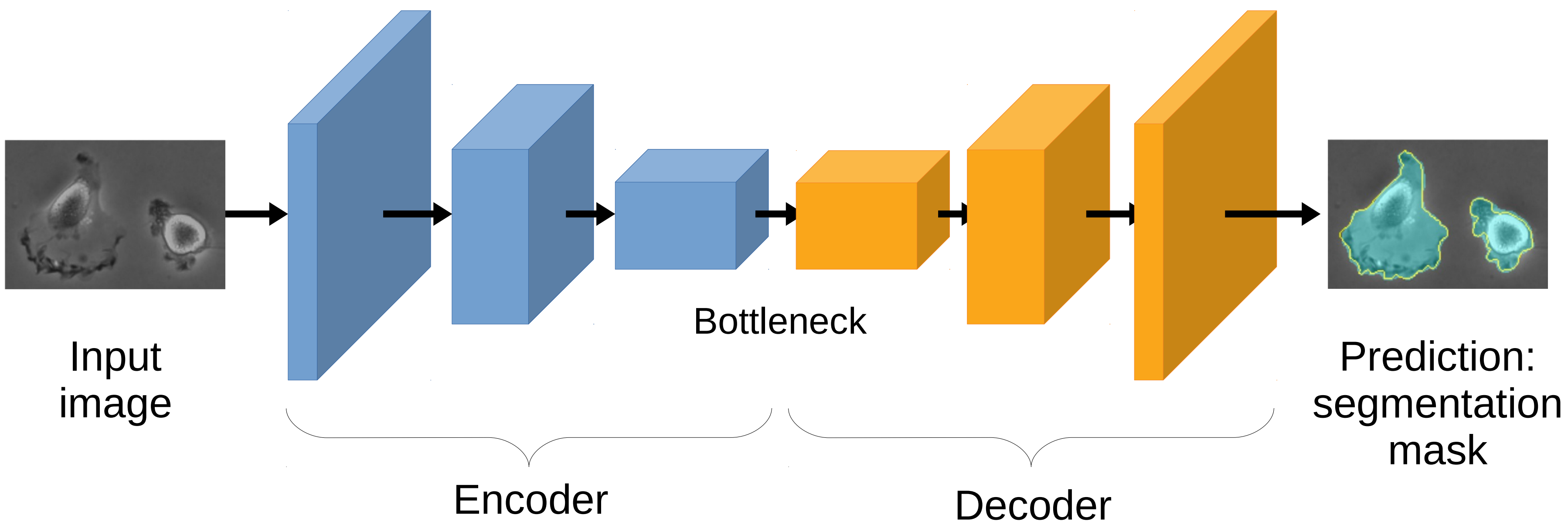
- Assign a class to each pixel
- Multiple classes with semantic meaning: person, dog, sheep, pig, background, ...

- Predict segmentation mask for foreground objects
- Instance specific (usually coupled with detection)

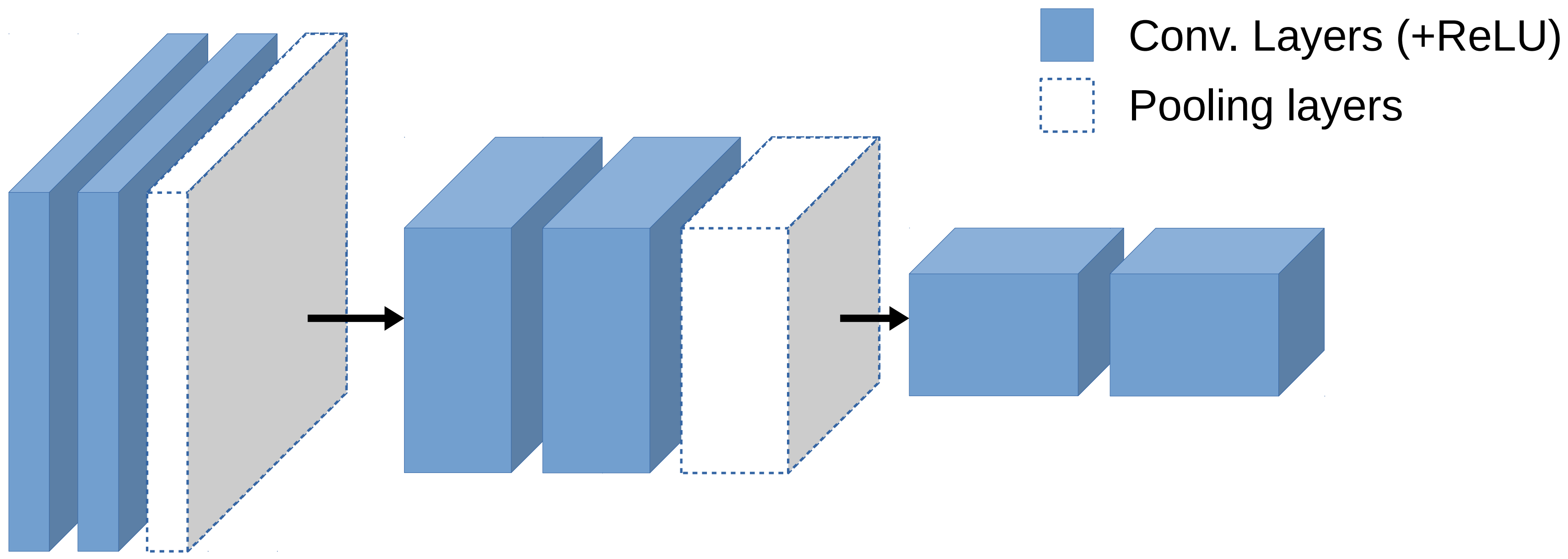
Sources: Ronneberger et al., *U-Net: Convolutional Networks for Biomedical Image Segmentation*, MICCAI 2015
He et al., *Mask R-CNN*, ICCV 2017

Segmentation with CNNs

Encoder-Decoder architecture:



Encoder Network



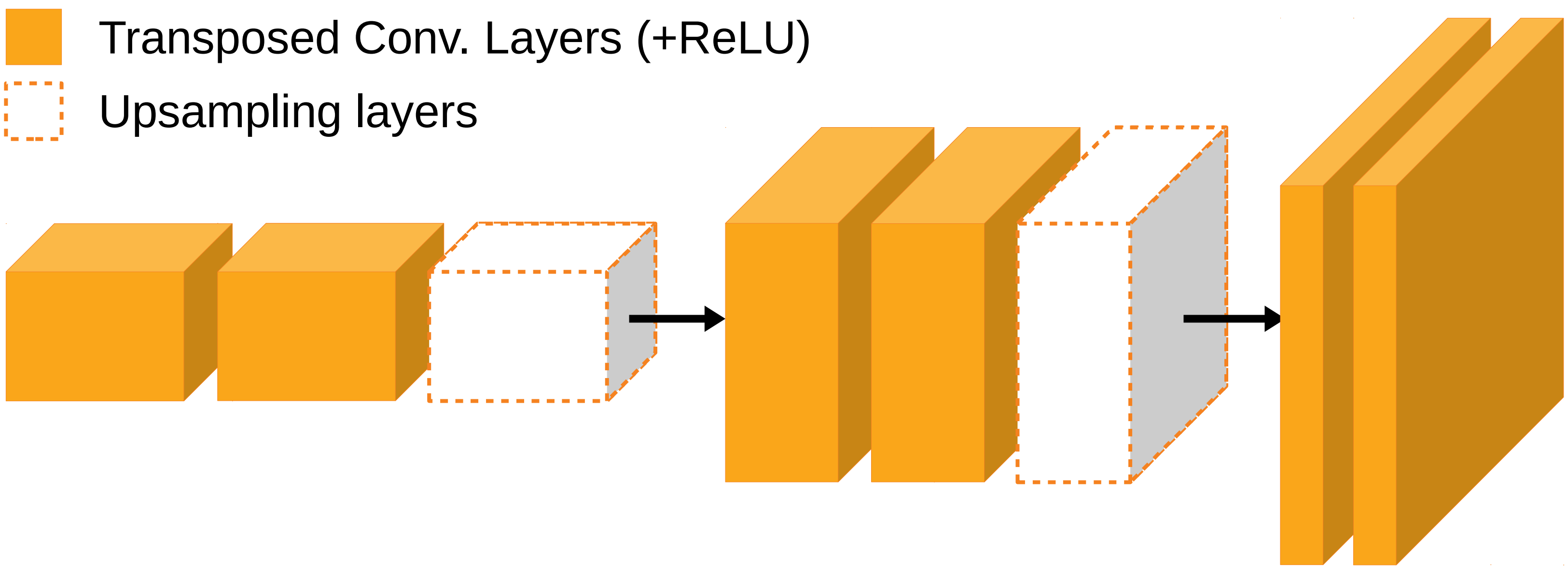
Input:

- Original resolution
- Low level representation (RGB)

Output/bottleneck:

- Low resolution
- High receptive field
- High level feature representation (high number of channels)

Decoder Network



- Uses feature representation to solve task
- Increases resolution via upsampling operations and/or transposed convolutions

Transposed Convolutions

- Also known as upconvolutions or deconvolutions
- They map the input to a higher resolution output
- Can be seen as “learned upsampling” operations

“Transposed” because:

a	b	c
d	e	f
g	h	i

convolutional filter

$$\downarrow$$

$$\mathbf{C} = \begin{pmatrix} a & b & c & 0 & d & e & f & 0 & g & h & i & 0 & 0 & 0 & 0 & 0 \\ 0 & a & b & c & 0 & d & e & f & 0 & g & h & i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a & b & c & 0 & d & e & f & 0 & g & h & i & 0 \\ 0 & 0 & 0 & 0 & 0 & a & b & c & 0 & d & e & f & 0 & g & h & i \end{pmatrix}$$

convolutional matrix

Transposed Convolutions

- Also known as upconvolutions or (wrongly) deconvolutions
- They map the input to a higher resolution output
- Can be seen as “learned upsampling” operations

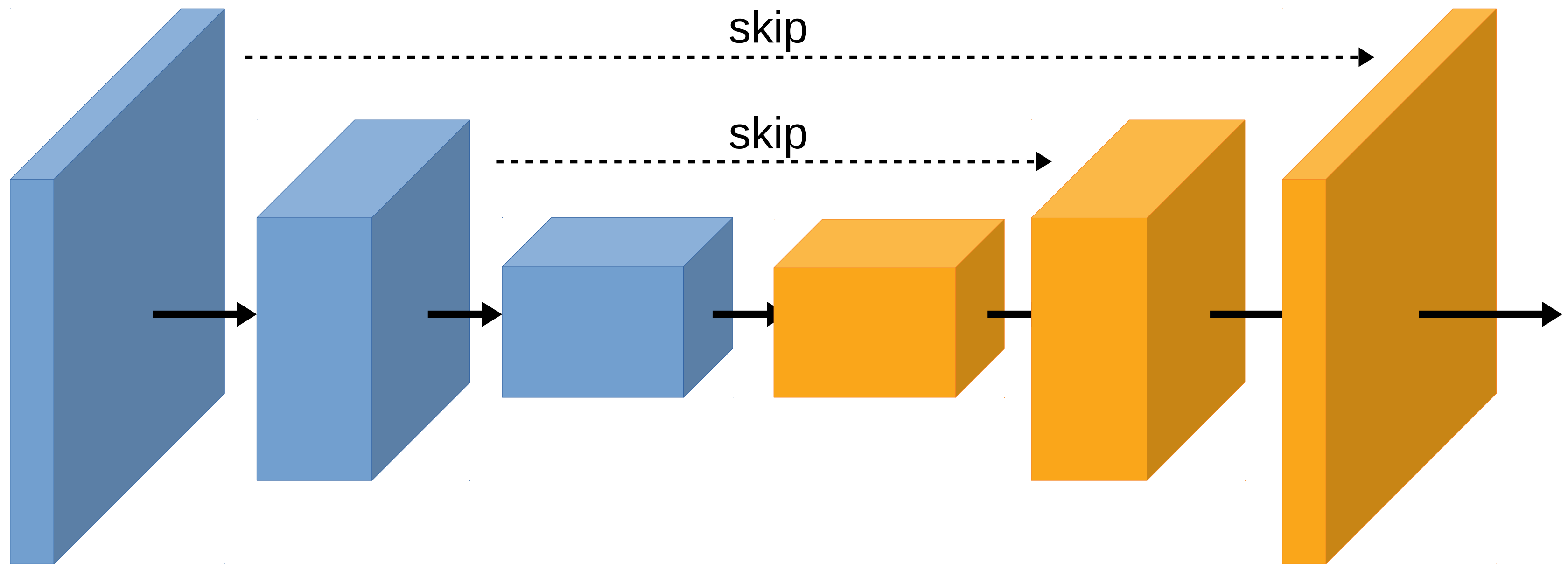
“Transposed” because:

Convolution: $$\mathbf{O} = \mathbf{C}\mathbf{I}$$

Transposed convolution: $$\mathbf{O} = \mathbf{C}^\top\mathbf{I}$$

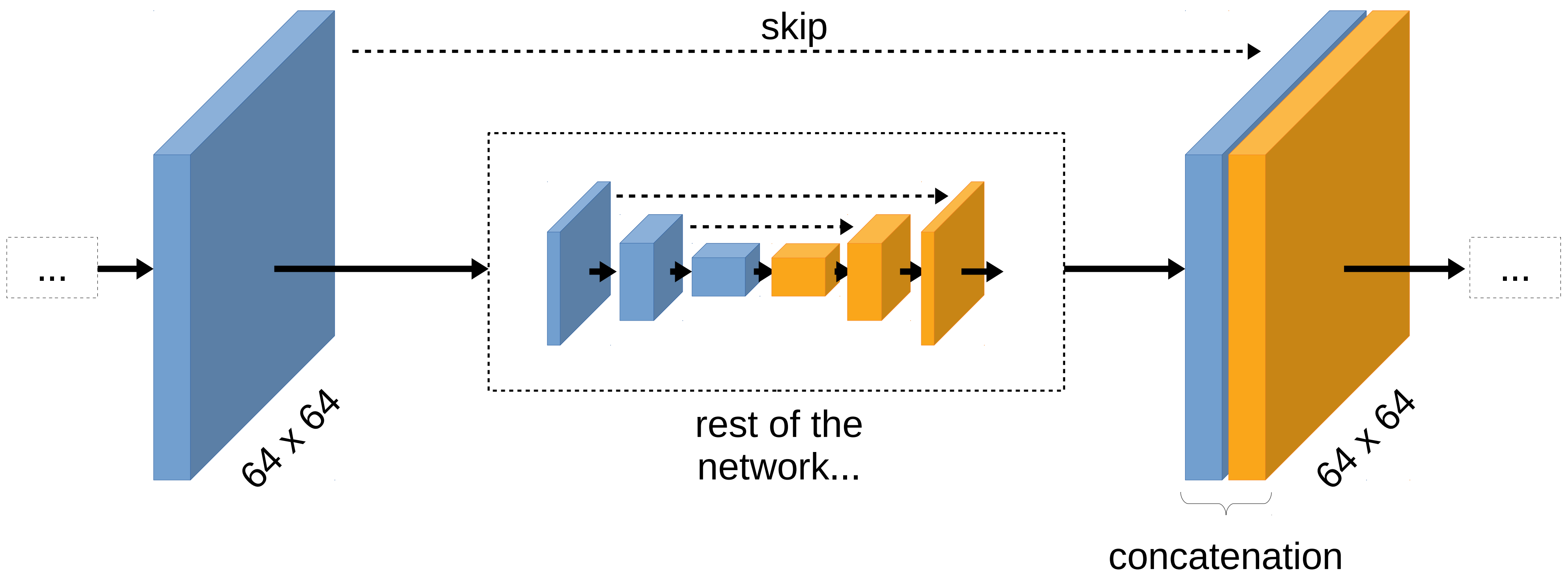
it can be computed using the transposed matrix of *some* convolution.

Skip Connections

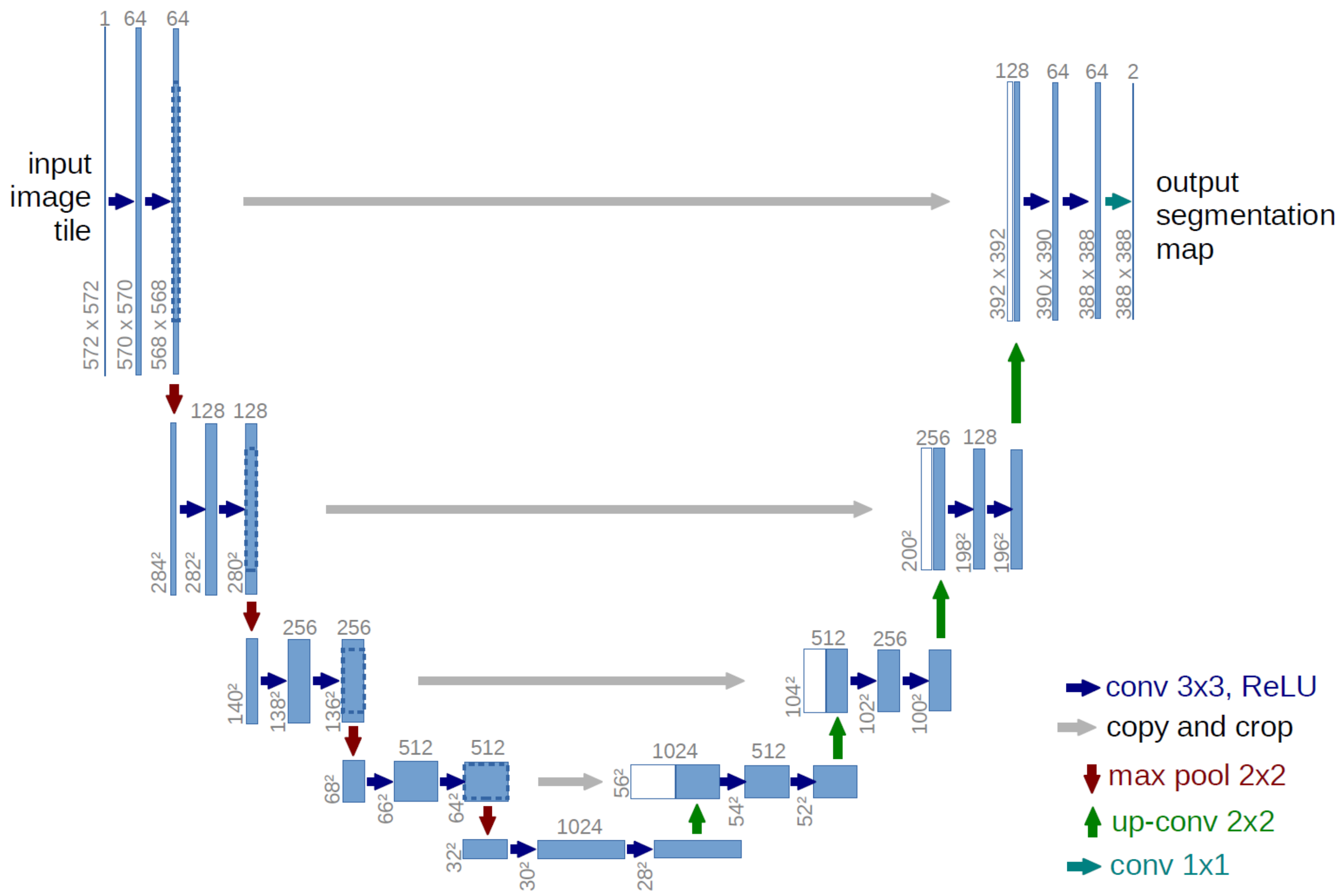


- “Shortcuts” from encoder activations to corresponding decoder stages
- Preserve high-res information, useful for refinement
- Improve sharpness of output

Skip Connections

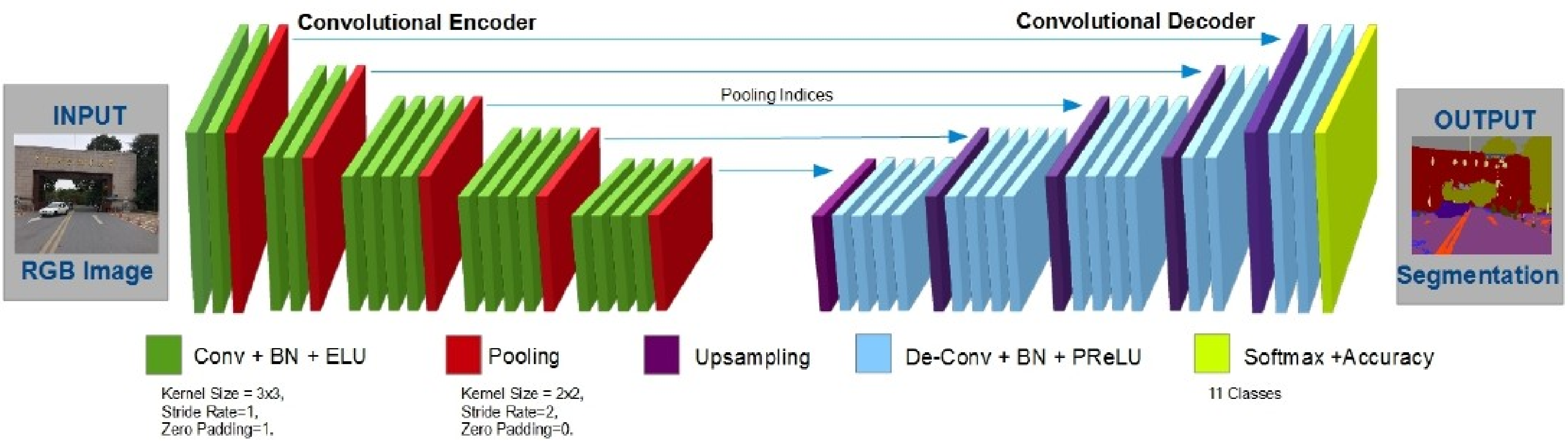


Example: U-Net



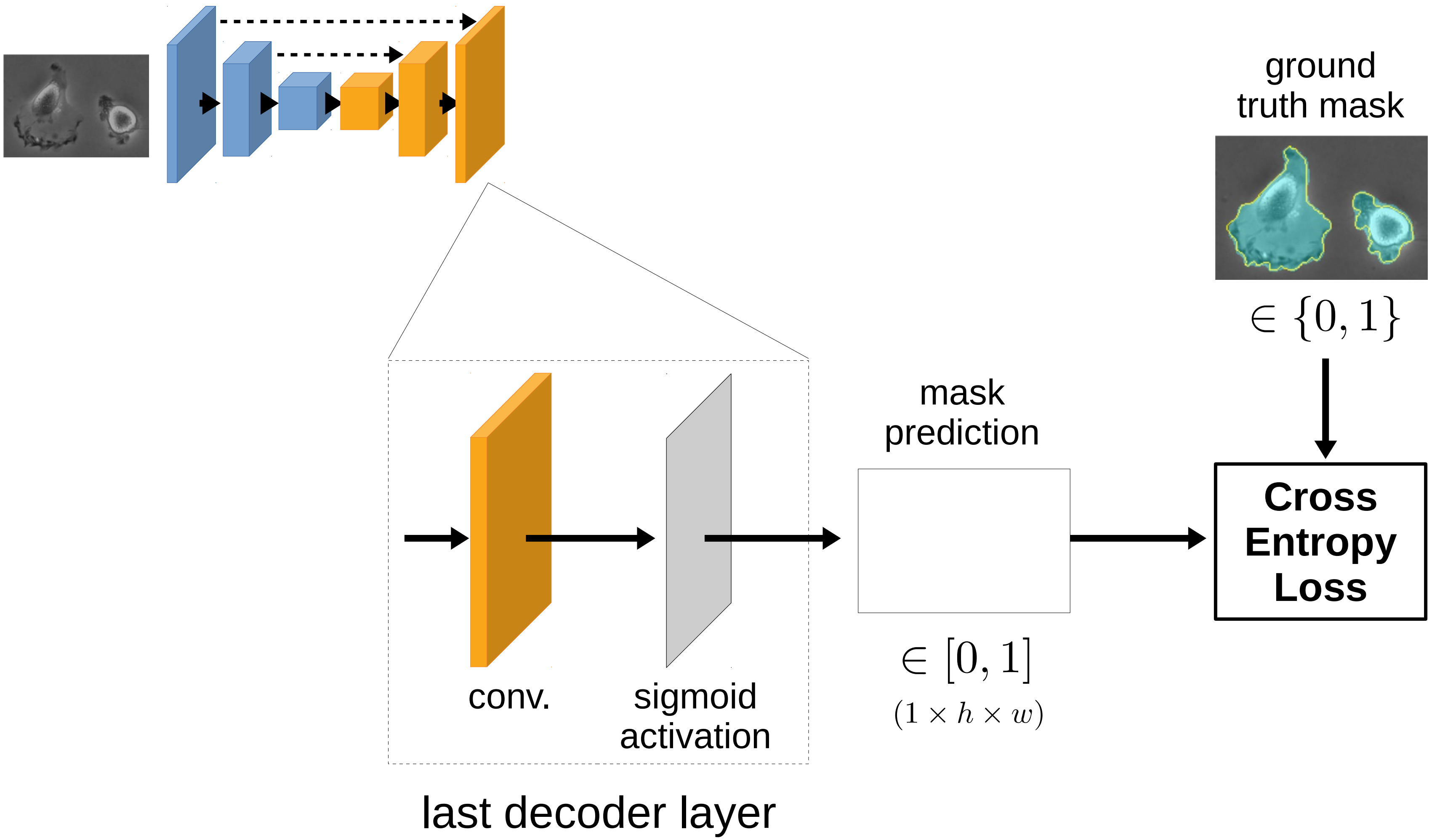
Ronneberger et al., *U-Net: Convolutional Networks for Biomedical Image Segmentation*, MICCAI 2015

Example: ECRU



Source: Robail Yasrab, "*ECRU: An Encoder-Decoder Based Convolution Neural Network (CNN) for Road-Scene Understanding",* Journal of Imaging 2018

Training for Segmentation

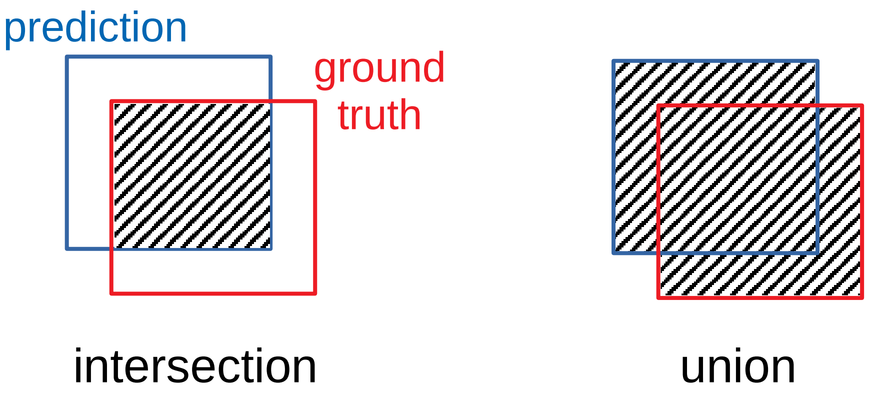


Evaluating Segmentation

Common evaluation metric for detection and segmentation:
Intersection over Union (IoU)

$$\mathrm{IoU} = \frac{\text{ground truth} \cap \text{prediction}}{\text{ground truth} \cup \text{prediction}}$$

E.g. in object detection:



A Few References

- Ronneberger et al., *"U-Net: Convolutional Networks for Biomedical Image Segmentation",* MICCAI 2015

- Robail Yasrab, *"ECRU: An Encoder-Decoder Based Convolution Neural Network (CNN) for Road-Scene Understanding", Journal of Imaging 2018*

- He et al., *"Mask R-CNN",* ICCV 2017

- He et al., *Deep Residual Learning for Image Recognition,* CVPR 2016

- Dumoulin et al., *A guide to convolution arithmetic for deep learning,* arXiv 1603.07285

Exercise

- Set up and train a model for pose estimation using the direct scalar regression approach.
- Implement the Softargmax loss and use it to train a second model. Compare it with the previous approach.
- Implement different encoder-decoder networks for segmentation and compare their performance.