Similarity learning

What can ML do for us?

- Classification problem

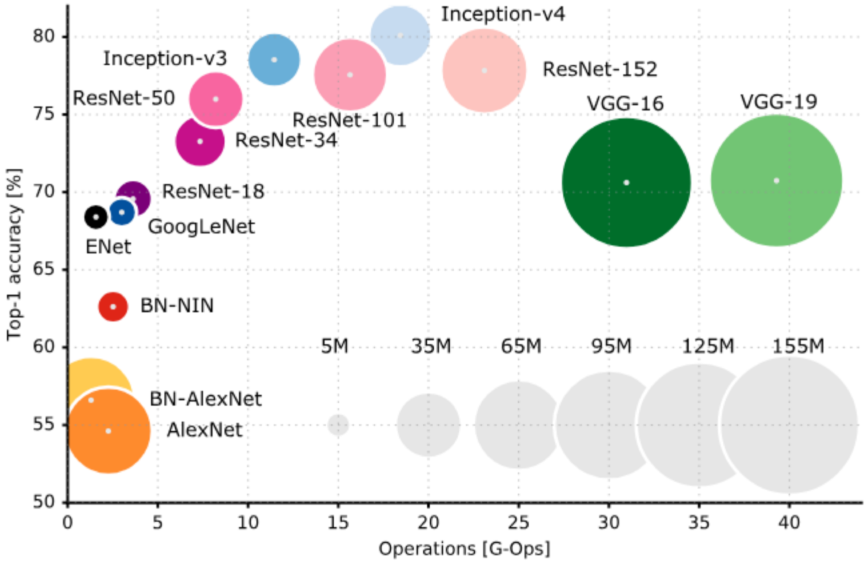
→ Neural Network → CAT

What can ML do for us?

- Classification problem on ImageNet with thousands of categories

What can ML do for us?

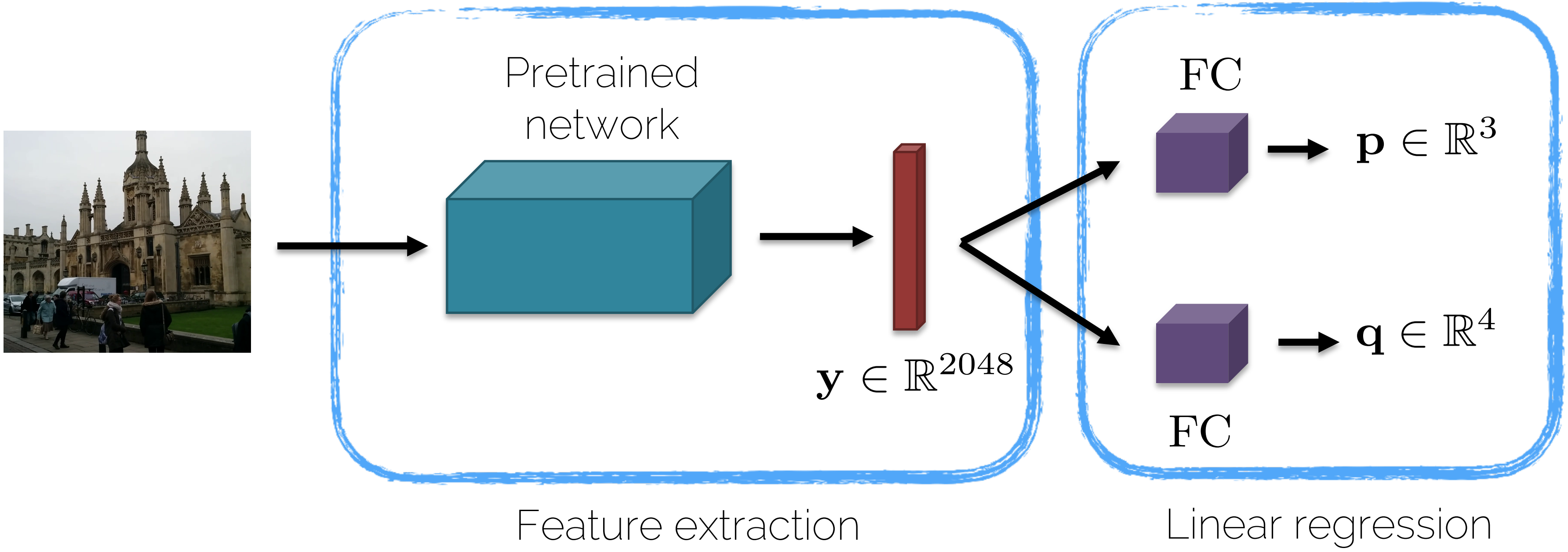
- Performance on ImageNet
  - Size of the blobs indicates the number of parameters



A. Canziani et al. „An Analysis of Deep Neural Network Models for Practical Applications". arXiv:1605.07678 2016

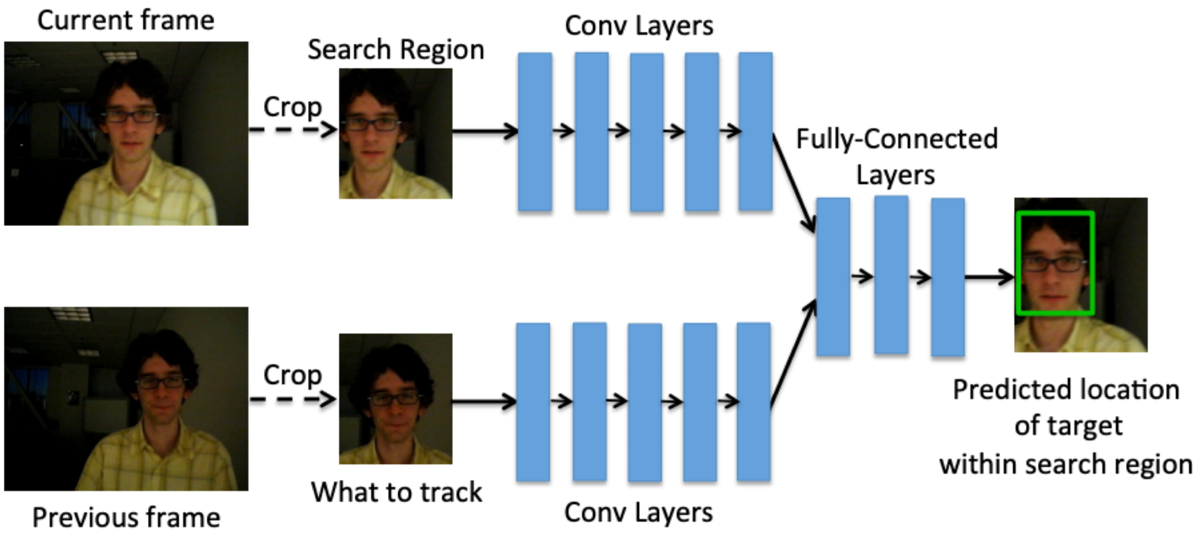
What can ML do for us?

- Regression problem: pose regression



What can ML do for us?

- Regression problem: bounding box regression



D. Held et al. „Learning to Track at 100 FPS with Deep Regression Networks". ECCV 2016

What can ML do for us?

- Third type of problems

A

Classification: person, face, female

B

Classification: person, face, male

What can ML do for us?

- Third type of problems

A

B

Is it the same person?

What can ML do for us?

- Third type of problems: Similarity Learning

A

B

- Comparison
- Ranking

Similarity Learning: when and why?

- Application: unlocking your iPhone with your face

Training

Similarity Learning: when and why?

- Application: unlocking your iPhone with your face

A  YES

B  NO

Testing 

Can be solved as a classification problem

Similarity Learning: when and why?

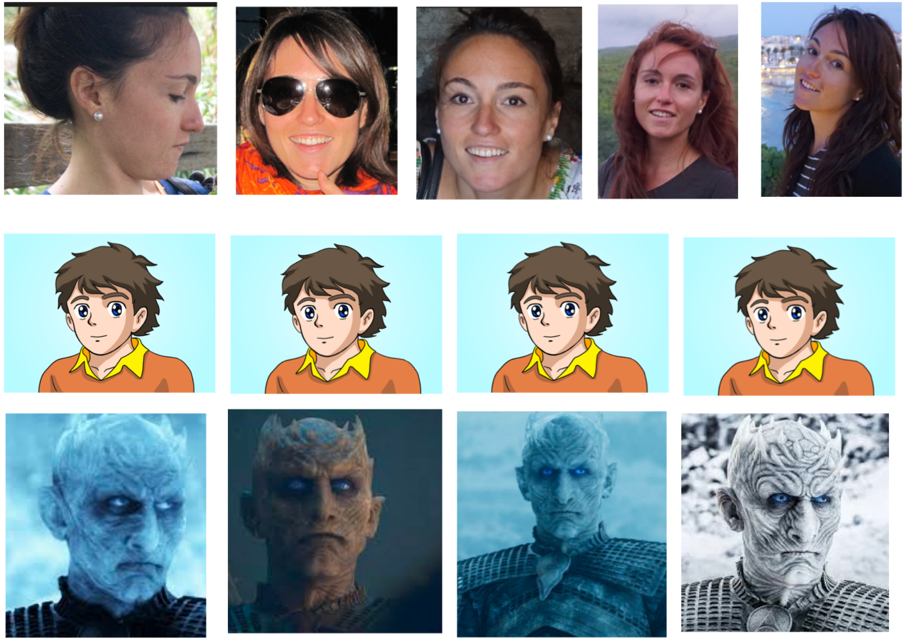
- Application: face recognition system so students can enter the exam room without the need for ID check

Training

Person 1

Person 2

Person 3



Similarity Learning: when and why?

- Application: face recognition system so students can enter the exam room without the need for ID check

What is the problem
with this approach?

Scalability – we need to retrain our model every time a new student is registered to the course

Similarity Learning: when and why?

- Application: face recognition system so students can enter the exam room without the need for ID check

Can we train one
model and use it every
year?

Similarity Learning: when and why?

- Learn a similarity function

A 

B 

Low similarity score

A 

B 

High similarity score

Similarity Learning: when and why?

- Learn a similarity function: testing

A

$$d(A, B) > \tau$$

Not the same person

B

Similarity Learning: when and why?

- Learn a similarity function

A

Same person

$$d(A, B) < \tau$$

B

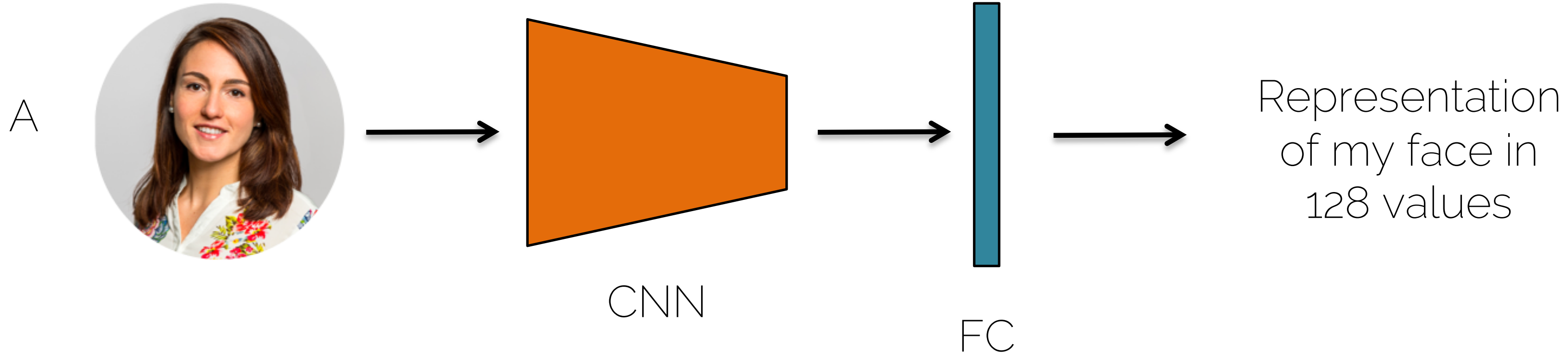
Similarity learning

- How do we train a network to learn similarity?

Siamese Neural Networks

Similarity learning

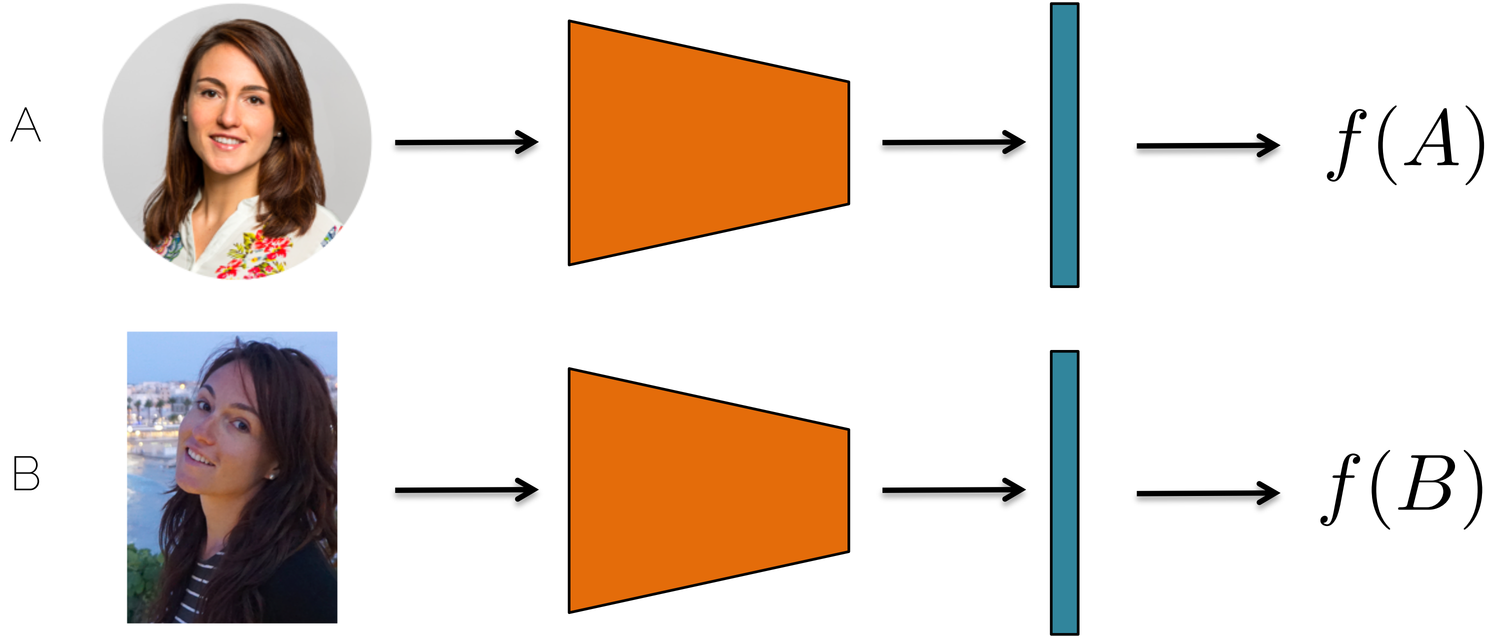
- How do we train a network to learn similarity?



Taigman et al. „DeepFace: closing the gap to human level performance". CVPR 2014

Similarity learning

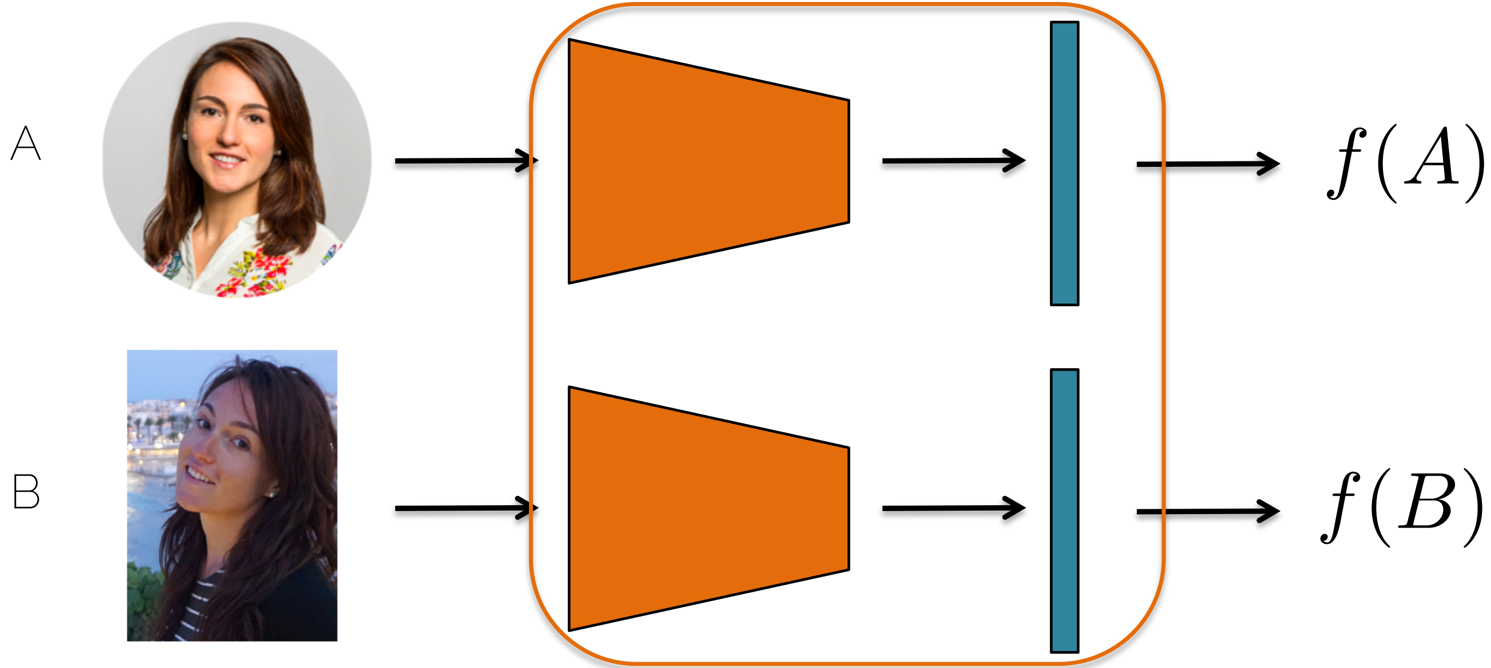
- How do we train a network to learn similarity?



Taigman et al. „DeepFace: closing the gap to human level performance“. CVPR 2014

Similarity learning

- Siamese network = shared weights



Taigman et al. „DeepFace: closing the gap to human level performance". CVPR 2014

Similarity learning

- Siamese network = shared weights
- We use the same network to obtain an encoding of the image $f(A)$
- To be done: compare the encodings

Taigman et al. „DeepFace: closing the gap to human level performance". CVPR 2014

Similarity learning

- Distance function $d(A, B) = ||f(A) - f(B)||^2$

- Training: learn the parameter such that

  - If A and B depict the same person, $d(A, B)$ is small

  - If A and B depict a different person, $d(A, B)$ is large

Taigman et al. „DeepFace: closing the gap to human level performance". CVPR 2014

Similarity learning

- Loss function for a positive pair:
  - If A and B depict the same person, $d(A, B)$ is small

$$\mathcal{L}(A, B) = ||f(A) - f(B)||^2$$

Similarity learning

- Loss function for a negative pair:
  - If A and B depict a different person, $d(A, B)$ is large
  - Better use a Hinge loss:

$$\mathcal{L}(A, B) = \max(0, m^2 - ||f(A) - f(B)||^2)$$

If two elements are already far away, do not spend energy in pulling them even further away

Similarity learning

- Contrastive loss:

$$\mathcal{L}(A,B) = y^*||f(A)-f(B)||^2 + (1-y^*)max(0, m^2 - ||f(A)-f(B)||^2)$$

Positive pair, reduce the distance between the elements

Negative pair, brings the elements further apart up to a margin

Similarity learning

- Training the siamese networks
  - You can update the weights for each channel independently and then average them

- This loss function allows us to learn to bring positive pairs together and negative pairs apart

Triplet loss

- Triplet loss allows us to learn a ranking

Anchor (A)

Positive (P)

Negative (N)

We want: $$||f(A) - f(P)||^2 < ||f(A) - f(N)||^2$$

Schroff et al „FaceNet: a unified embedding for face recognition and clustering". CVPR 2015

Triplet loss

- Triplet loss allows us to learn a ranking

$$||f(A) - f(P)||^2 < ||f(A) - f(N)||^2$$

$$||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 < 0$$

$$||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + m < 0$$

margin

Schroff et al „FaceNet: a unified embedding for face recognition and clustering". CVPR 2015

Triplet loss

- Triplet loss allows us to learn a ranking

$$||f(A) - f(P)||^2 < ||f(A) - f(N)||^2$$

$$||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 < 0$$

$$||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + m < 0$$

$$\mathcal{L}(A, P, N) = max(0, ||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + m)$$

Schroff et al „FaceNet: a unified embedding for face recognition and clustering". CVPR 2015

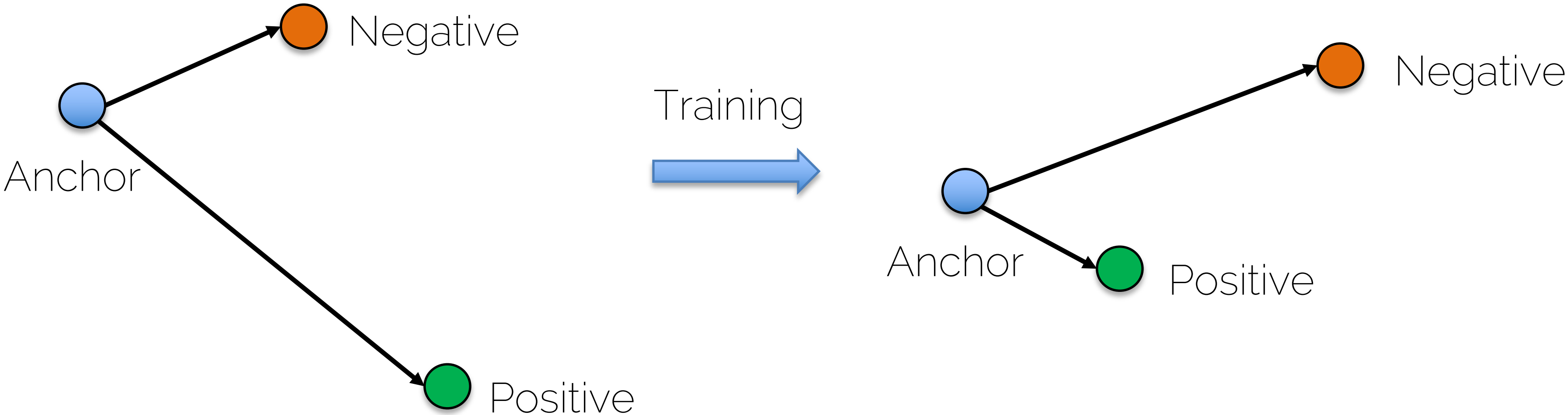
Triplet loss

- Training with hard cases

$$\mathcal{L}(A,P,N) = max(0, ||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + m)$$

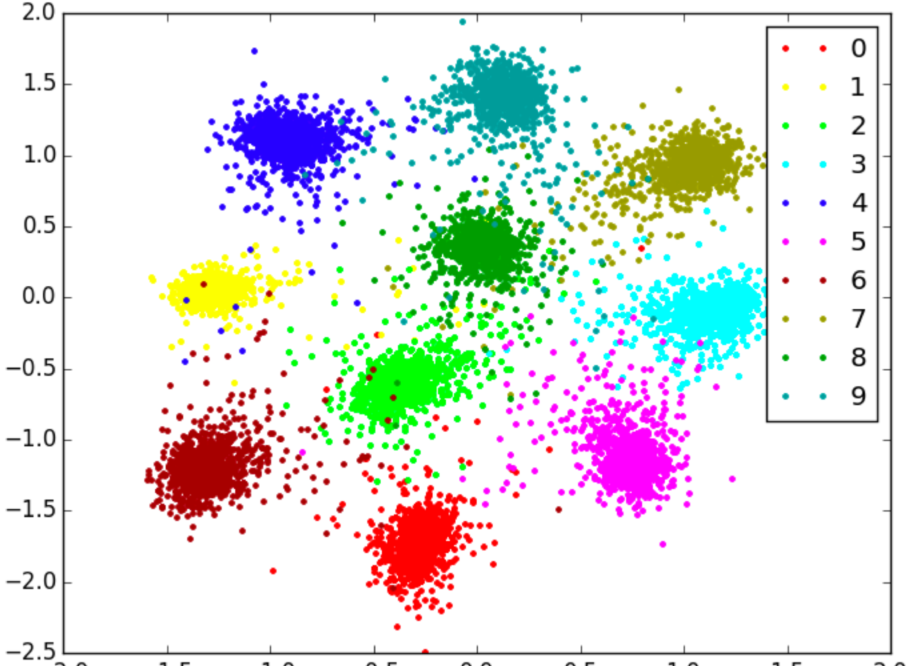
- Train for a few epochs
- Choose the hard cases where $d(A,P) \approx d(A,N)$
- Train with those to refine the distance learned

Triplet loss



Applications in vision

Siamese network on MNIST



Establishing image correspondences

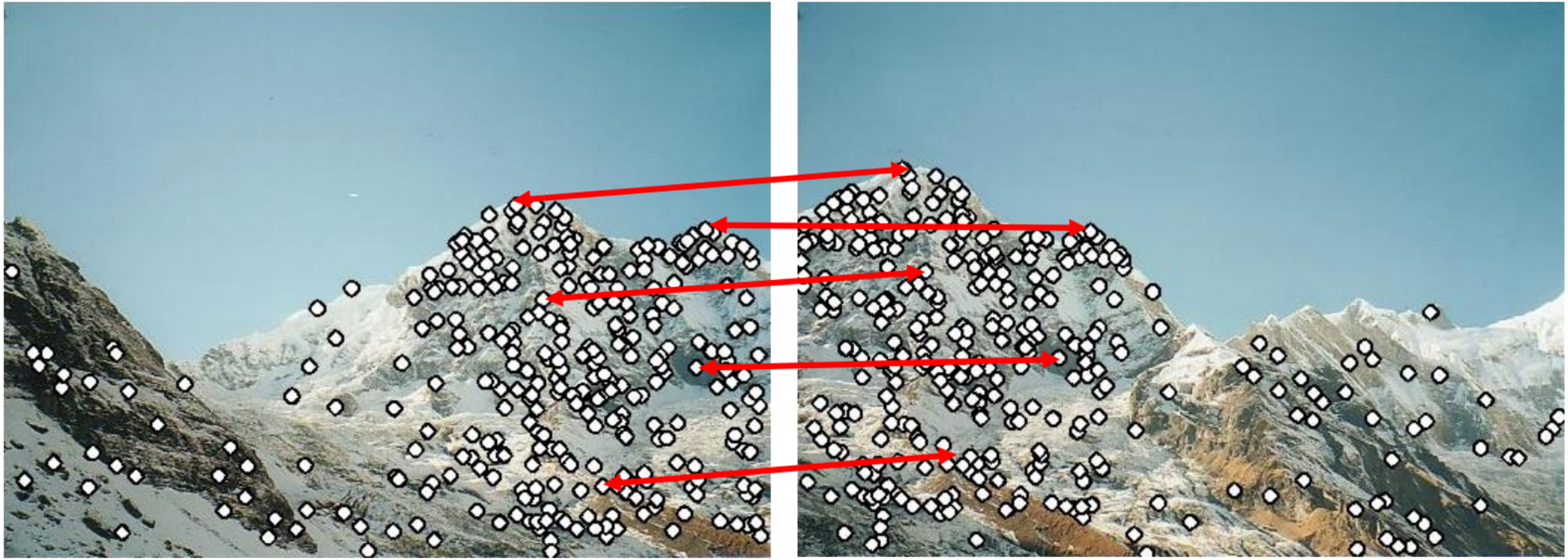


Image from University of Washington

Establishing image correspondences

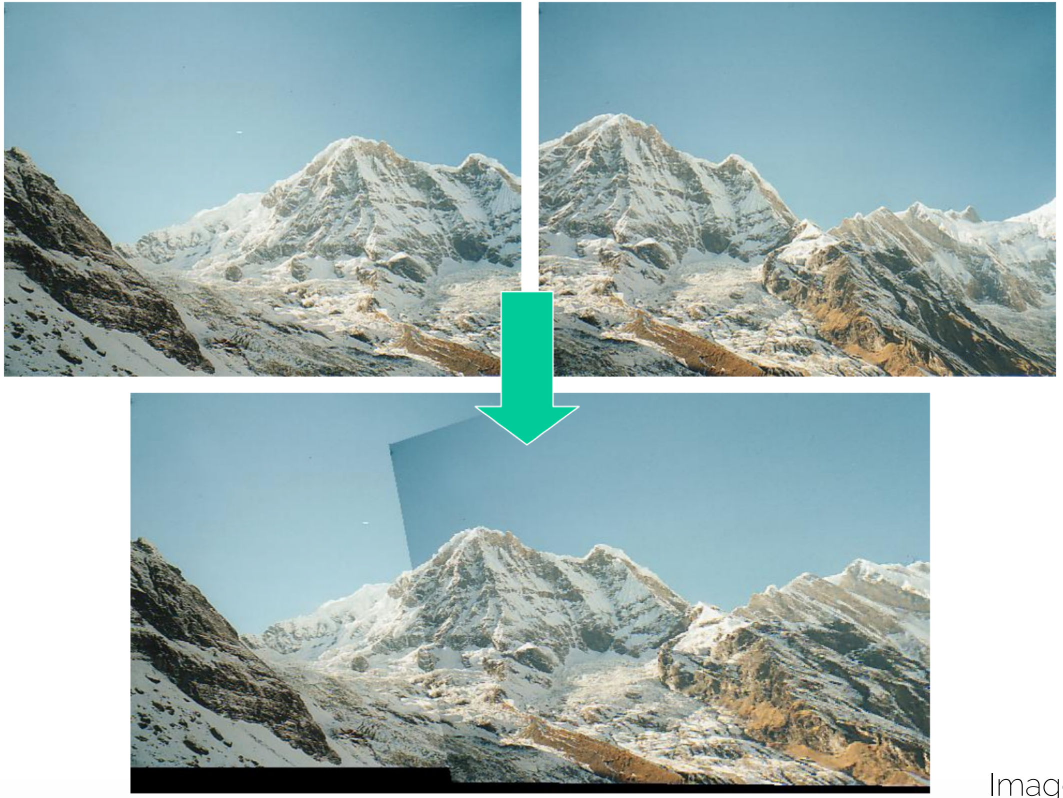


Image from University of Washington

Establishing image correspondences

- Used in a wide range of Computer Vision applications
  - Image stitching or image alignment
  - Object recognition
  - 3D reconstruction
  - Object tracking
  - Image retrieval

- Many of these applications are now targeted directly with Neural Networks as we will see in the course

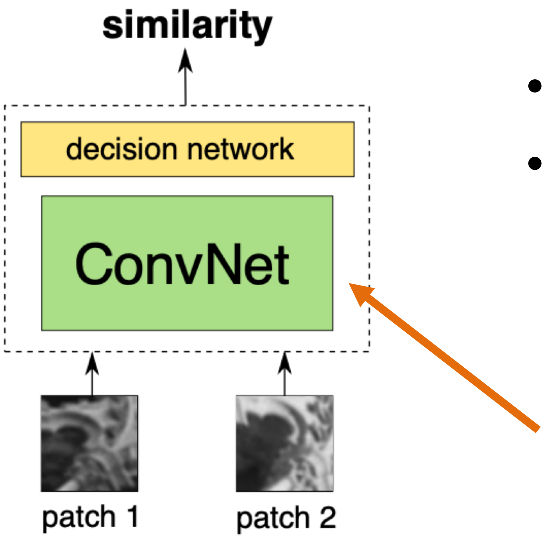
Establishing image correspondences

- Classic method pipeline
  - Extract manually designed feature descriptors
    - Harris, SIFT, SURF: most are based on image gradients
    - They suffer under extreme illumination or viewpoint changes
    - Slow to extract dense features
  - Match descriptors from the two images
    - Many descriptors are similar, one needs to filter out possible double matches and keep only reliable ones.

Sameer Agarwal et al. „Building Rome in a Day". ICCV 2009

Establishing image correspondences

- End-to-end learning for patch similarity



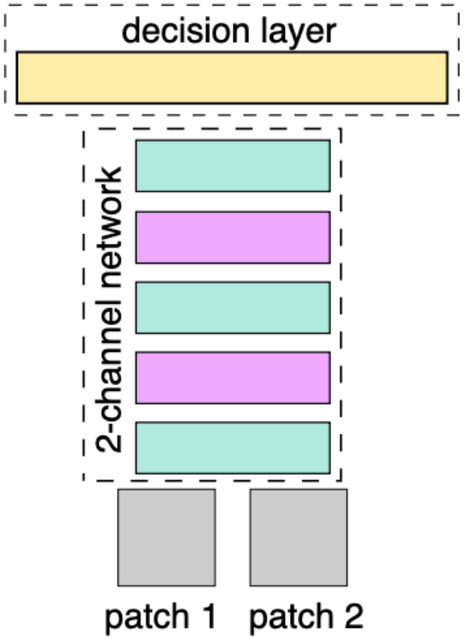
- Fast to allow dense extraction
- Invariant to a wide array of transformations (illumination, viewpoint)

Siamese network

S. Zagoruyko and N. Komodakis. „Learning to Compare Image Patches via Convolutional Neural Networks". CVPR 2015

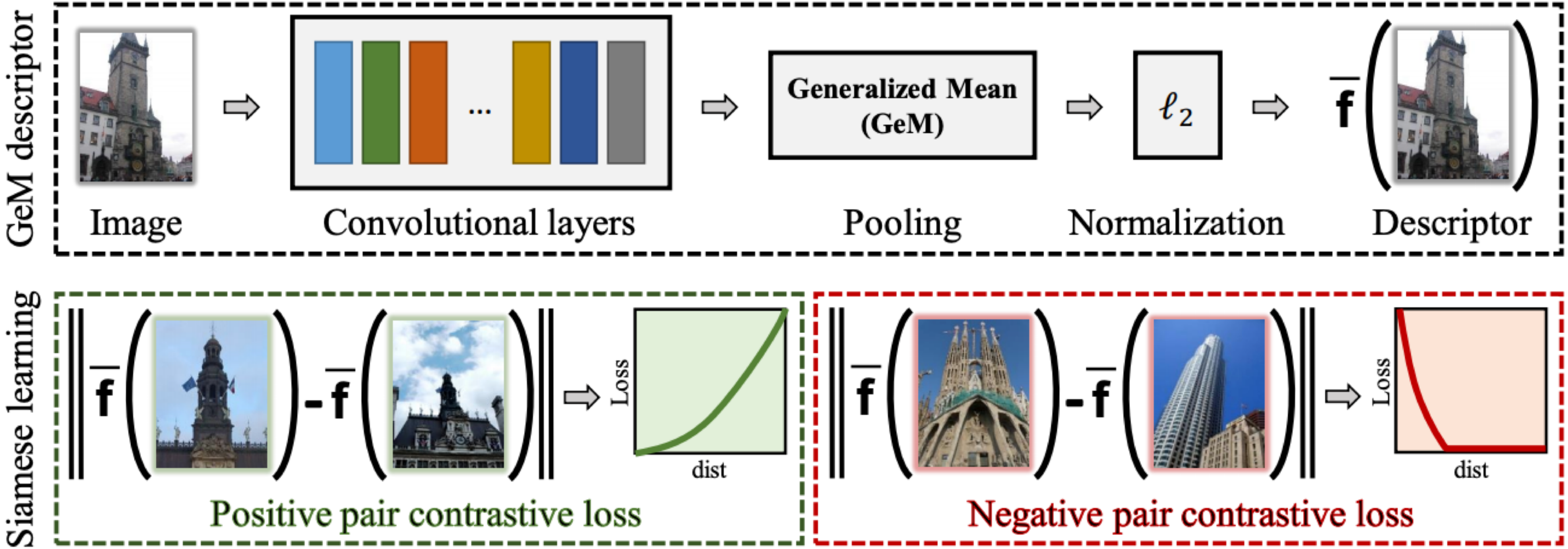
Establishing image correspondences

- Classic Siamese architecture
  - Shared layers
    - Simulated feature extraction
  - One decision layer
    - Simulates the matching



S. Zagoruyko and N. Komodakis. „Learning to Compare Image Patches via Convolutional Neural Networks“. CVPR 2015

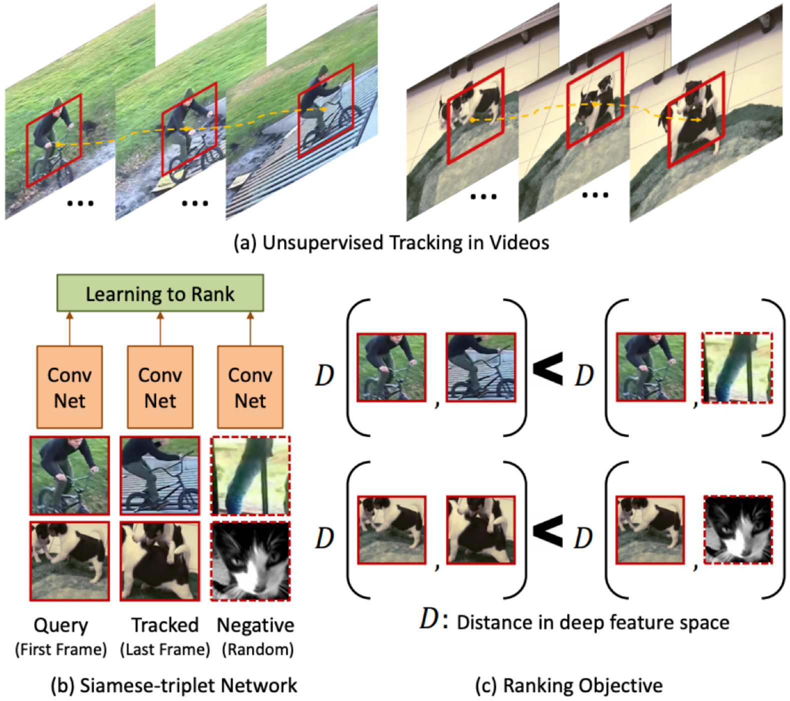
Image retrieval



Radenovic et al.. „Fine-tuning CNN Image Retrieval with No Human Annotation". TPAMI 2018

Unsupervised learning

- Learning from videos
  - Tracking provides the supervision
  - Use those as positive samples
  - Extract random patches as negative samples



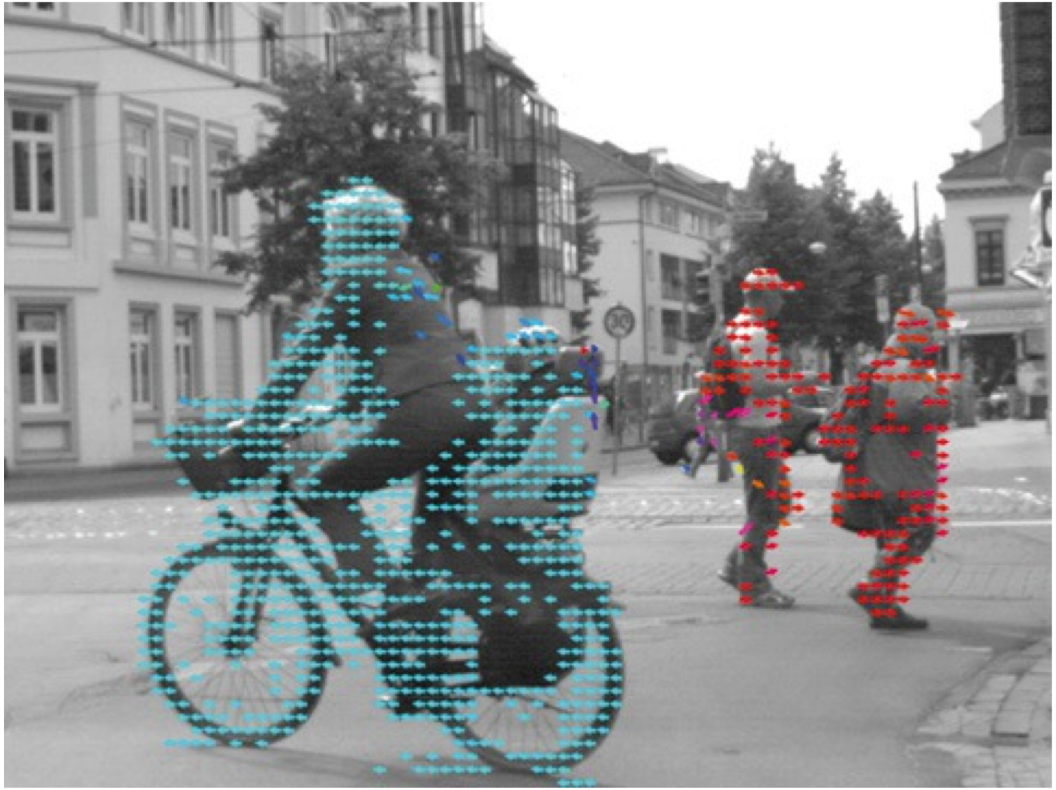
Wang and Gupta. „Unsupervised Learning of Visual Representations using Videos". ICCV 2015

Optical flow

- Input: 2 consecutive images (e.g. from a video)
- Output: displacement of every pixel from image A to image B

- Results in the “perceived” 2D motion, not the real motion of the object

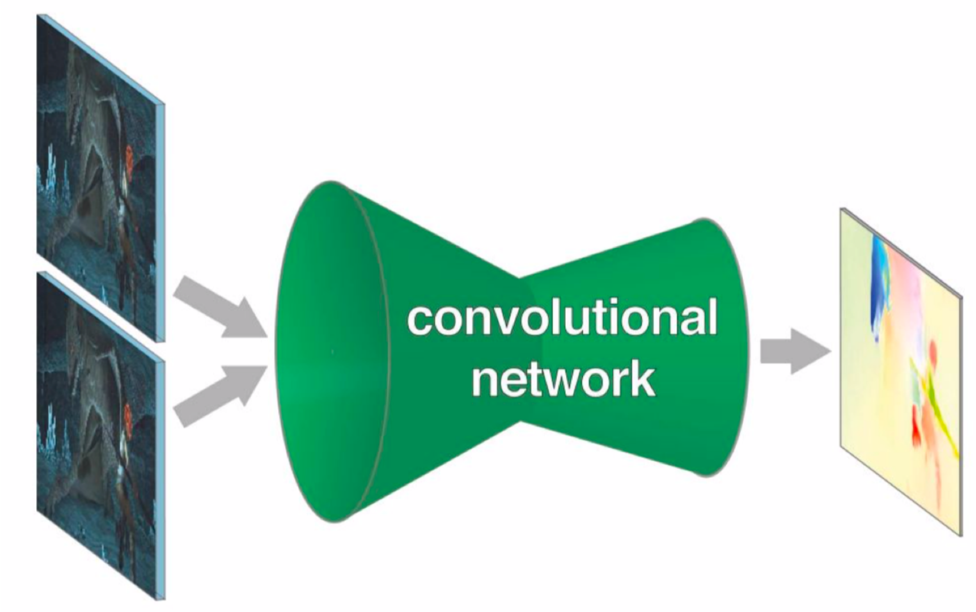
Optical flow



Optical flow

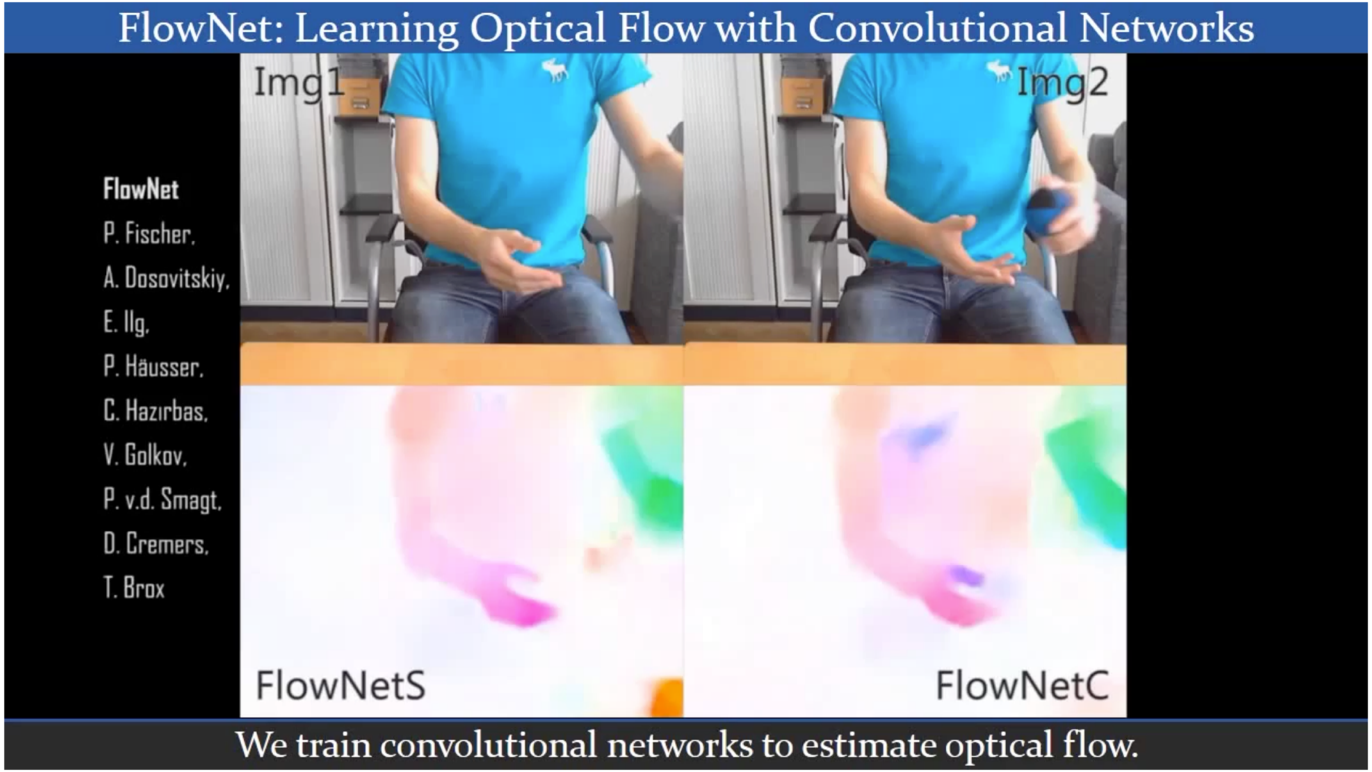
Optical flow with CNNs

- End-to-end supervised learning of optical flow



P. Fischer et al. „FlowNet: Learning Optical Flow With Convolutional Networks“. ICCV 2015

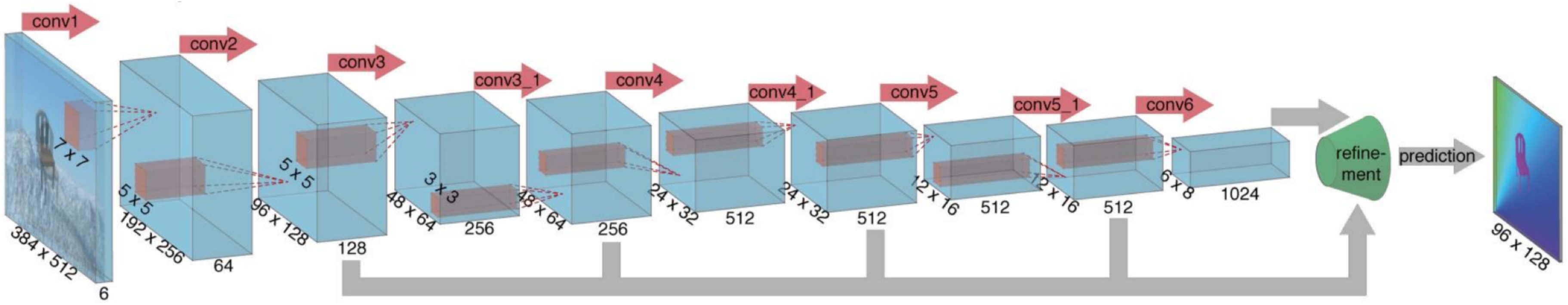
Optical flow with CNNs



P. Fischer et al. „FlowNet: Learning Optical Flow With Convolutional Networks". ICCV 2015

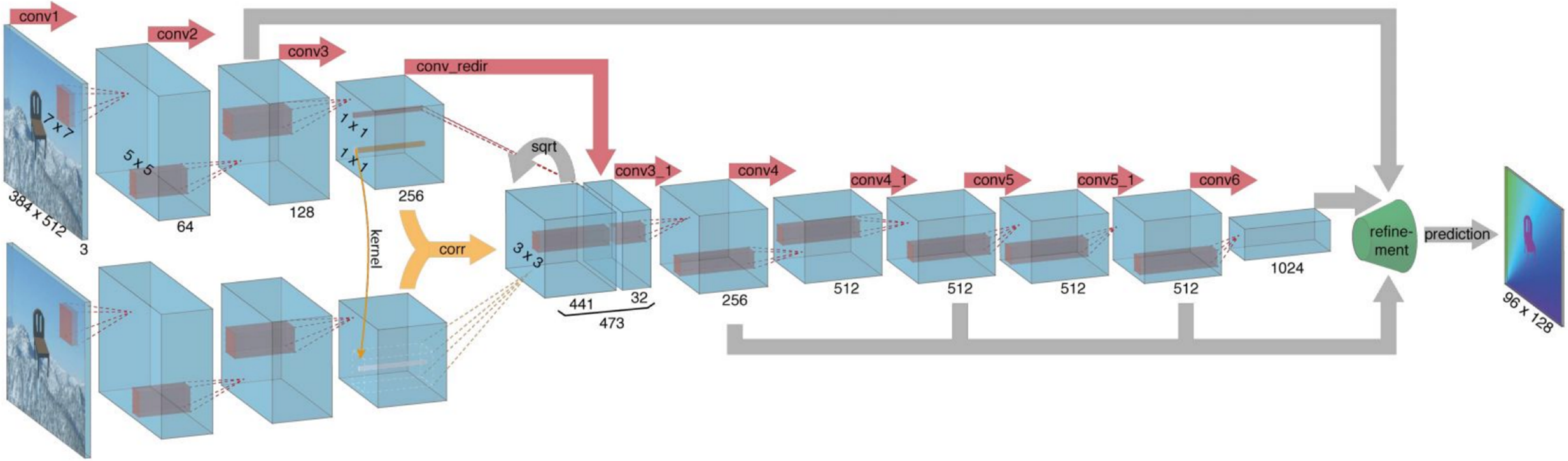
FlowNet: architecture 1

- Stack both images → input is now 2 x RGB = 6 channels



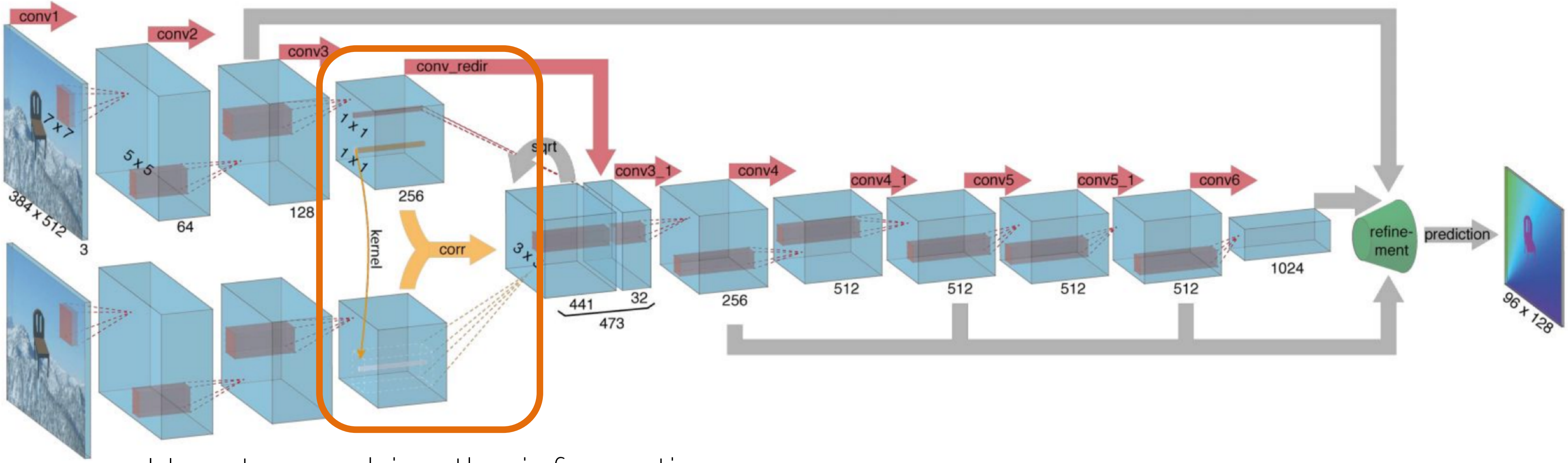
FlowNet: architecture 2

- Siamese architecture



FlowNet : architecture 2

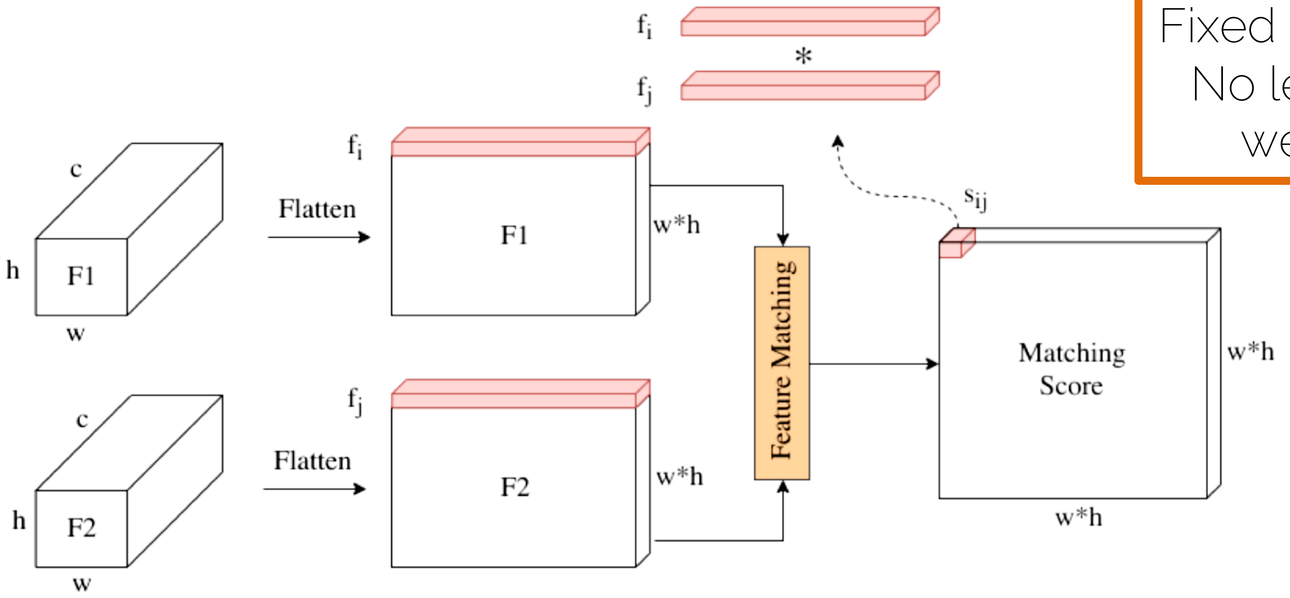
- Two key design choices



How to combine the information from both images?

Correlation layer

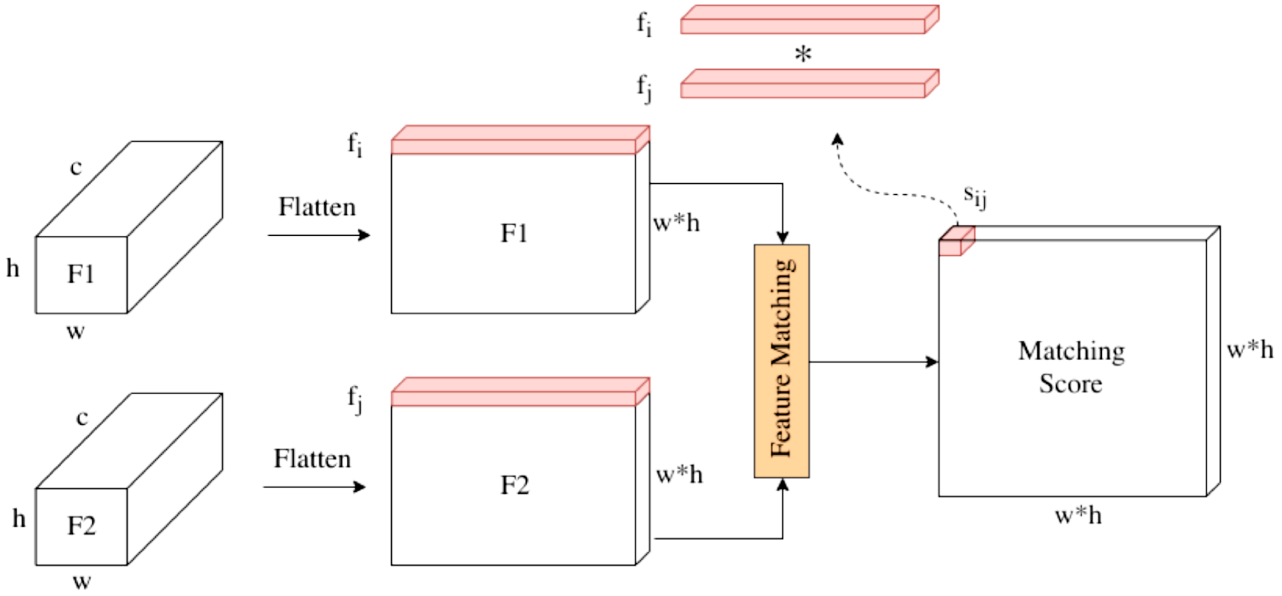
- Multiplies a feature vector with another feature vector



Fixed operation.
No learnable weights!

Correlation layer

- The matching score represents how correlated these two feature vectors are



Correlation layer

- Useful for finding image correspondences

Find a transformation from image A to image B

I. Rocco et al. "Convolutional neural network architecture for geometric matching. CVPR 2017.

Correlation layer

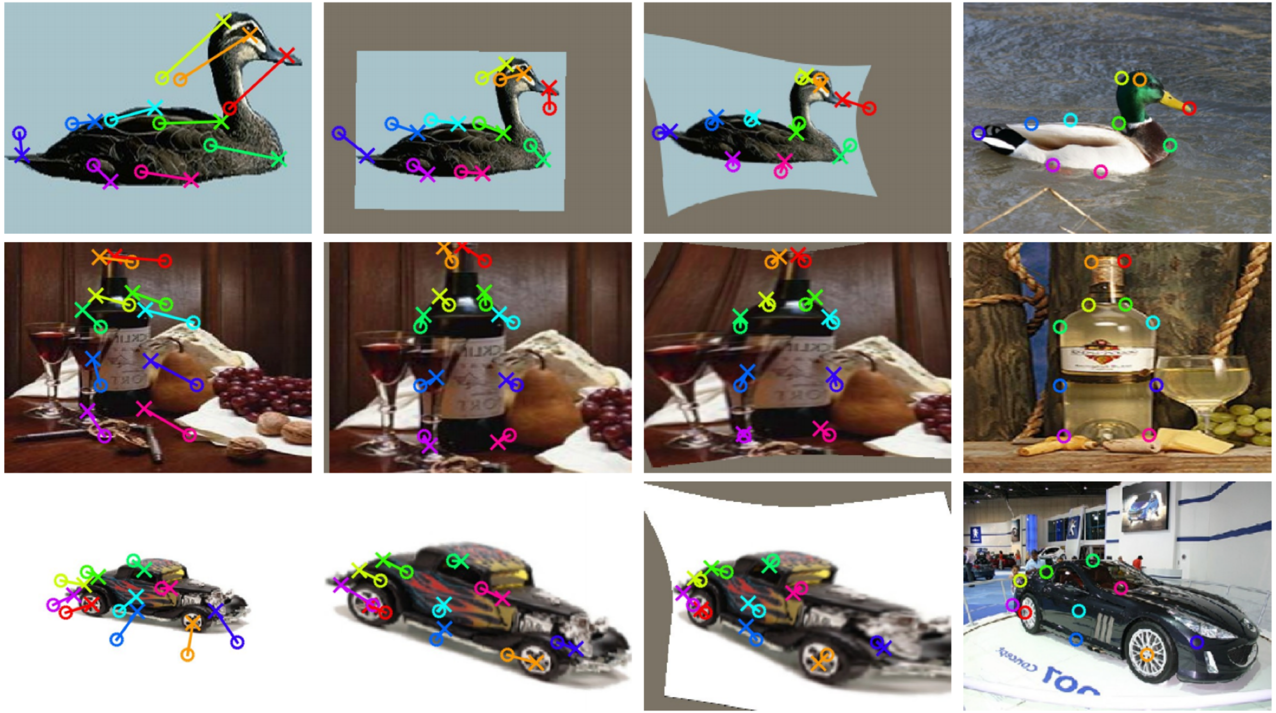
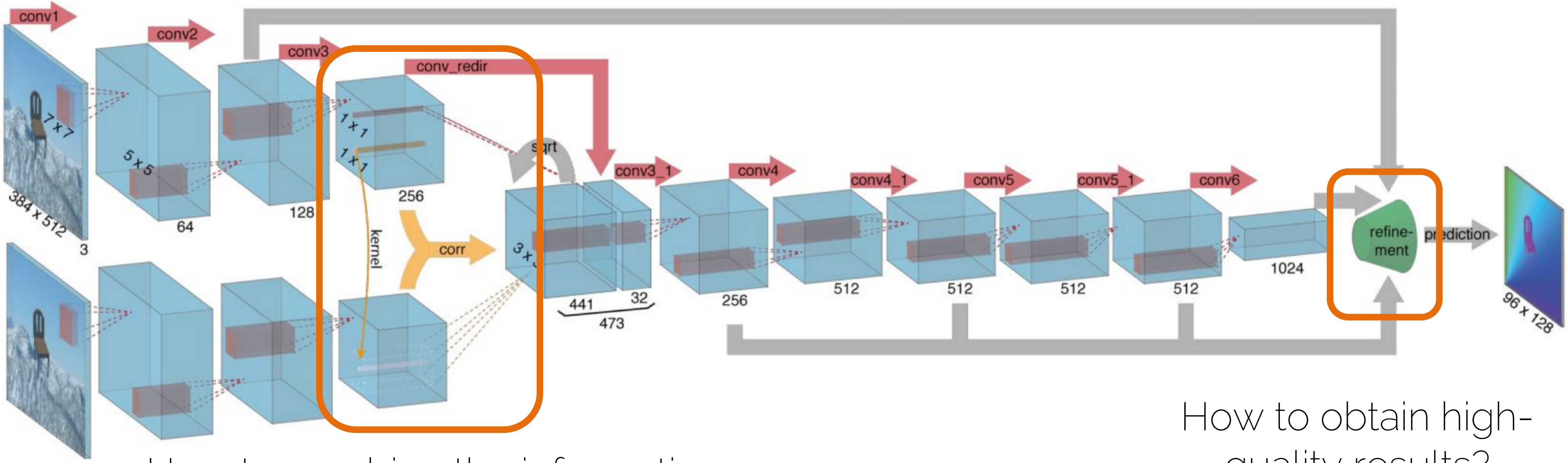


Image A | Aligned A (affine) | Aligned A (affine+TPS) | Image B

I. Rocco et al. "Convolutional neural network architecture for geometric matching. CVPR 2017.

FlowNet : architecture 2

- Two key design choices



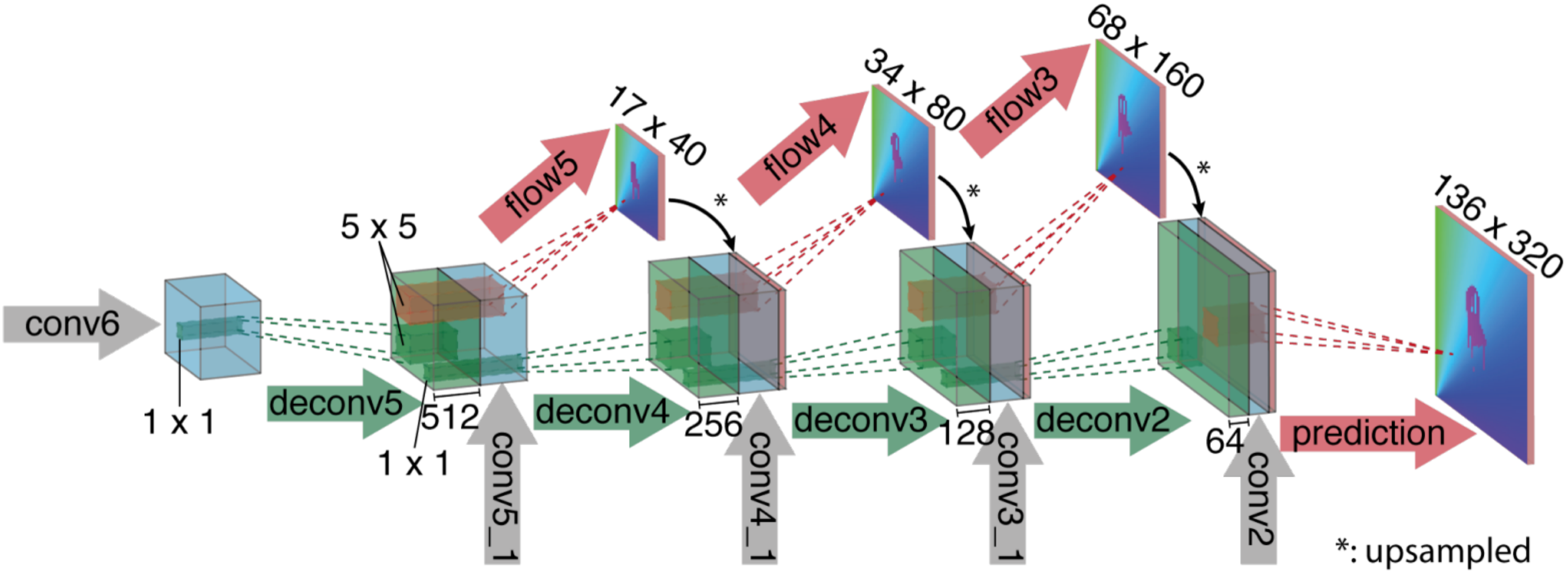
How to combine the information from both images?

How to obtain high-quality results?

FlowNet : architecture 2

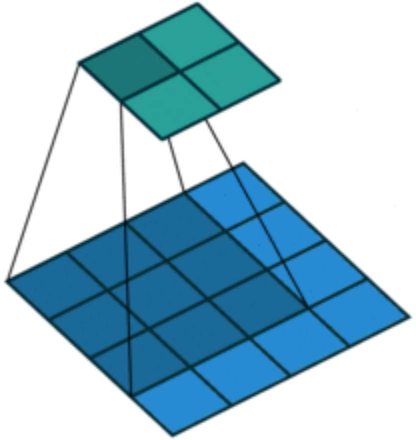
- Convolutions + pooling are great to allow aggregation of information from different parts of the image
- It also makes computation feasible!
- Problem: it reduces the size of our input, if we want full sized outputs (segmentation, optical flow) we need further operations

Refinement architecture

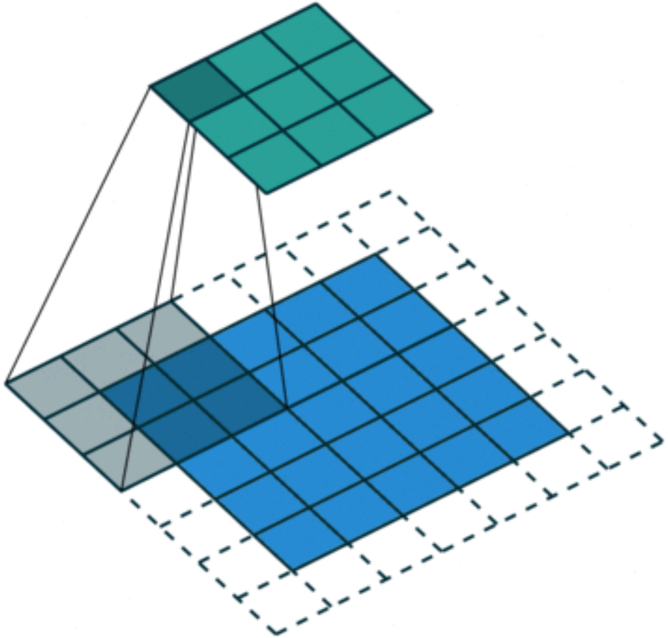


Convolution

- Recall



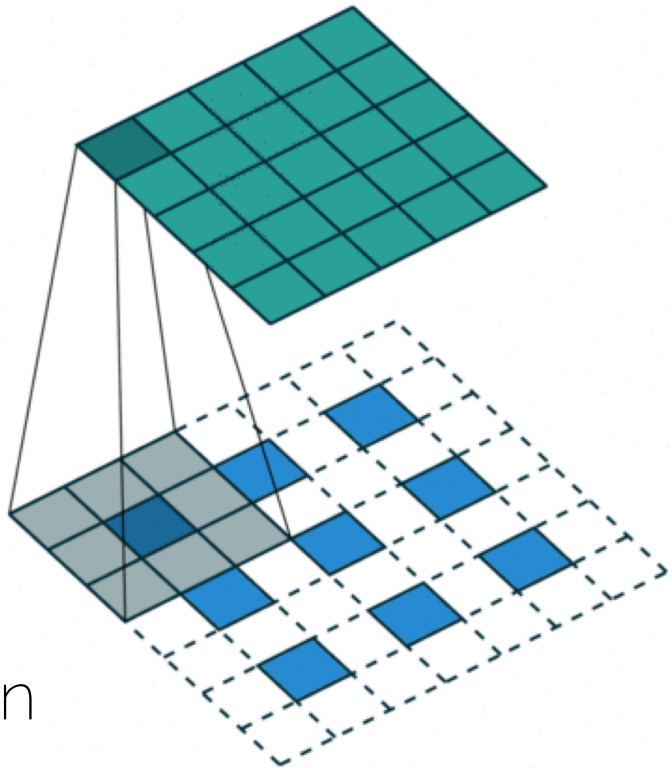
Convolution
no padding, no stride



Convolution
padding, stride

Transpose convolution

- We want to convert the 3x3 input into a 5x5 output
- Clever padding on the input plus a normal convolution
- Unpooling + conv = upconvolution



More on that later

- Next step: Autoencoder architecture as to generate outputs of the same size as inputs

Cool things you can do

- Savinov et al. „Quad-networks: unsupervised learning to rank for interest point detection“. CVPR 2017
- Ristani & Tomasi. „Features for Multi-Target Multi-Camera Tracking and Re-Identification“. CVPR 2018
- Chen et al. „Beyond triplet loss: a deep quadruplet network for person re-identification“. CVPR 2017

Next lecture

- No practical session on Wednesday
- We will send you the proposal feedback this week
- Next Monday: more on advanced architectures (attention and conditioning)