

Vision and Perception

Deep Learning for Video: Motion estimation



SAPIENZA
UNIVERSITÀ DI ROMA

References

- Basic reading: Szeliski, Chapter 9.3

Optical Flow

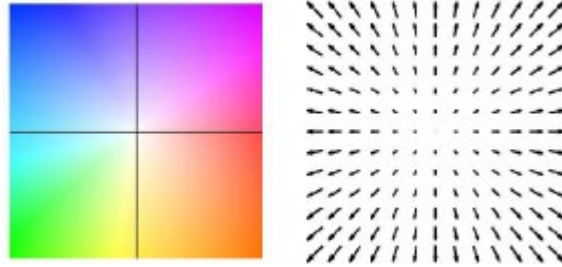
- Motion estimation in video
 - “Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image.”¹
 - For subsequent frames, determine displacement vector for each pixel
-
- Possible applications: visual odometry, autonomous driving, semantic segmentation...
→ Whenever motion conveys useful information



¹ Horn, Berthold KP, And Brian G. Schunck. "Determining Optical Flow." Artificial Intelligence 17.1-3 (1981): 185-203. <https://devblogs.nvidia.com/an-introduction-to-the-nvidia-optical-flow-sdk/>, retrieved 18.11.2019

Color code visualization

- Colour code for visualisation:



Optical flow with deep neural networks

- “Classical” approaches: complex optimization problems, computationally expensive

→ Not suitable for real-time applications



- First DNN approaches: trade-off between accuracy and size of the model

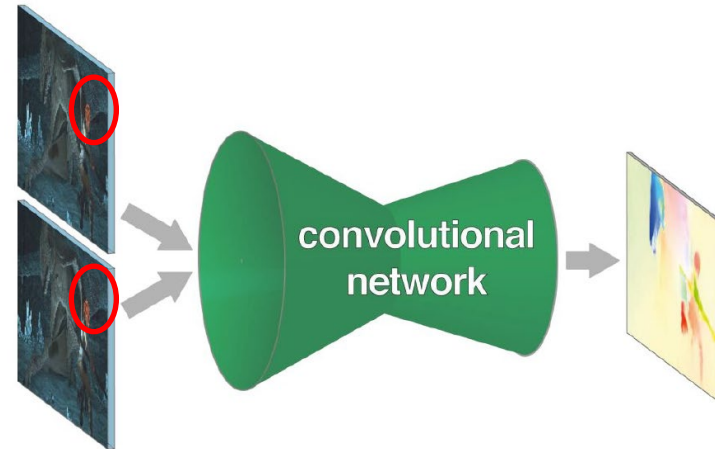
FlowNet

End to end supervised learning for optical flow estimation:

- encoder/decoder architecture

The idea: given two consecutive frames need to match features at different location in input images

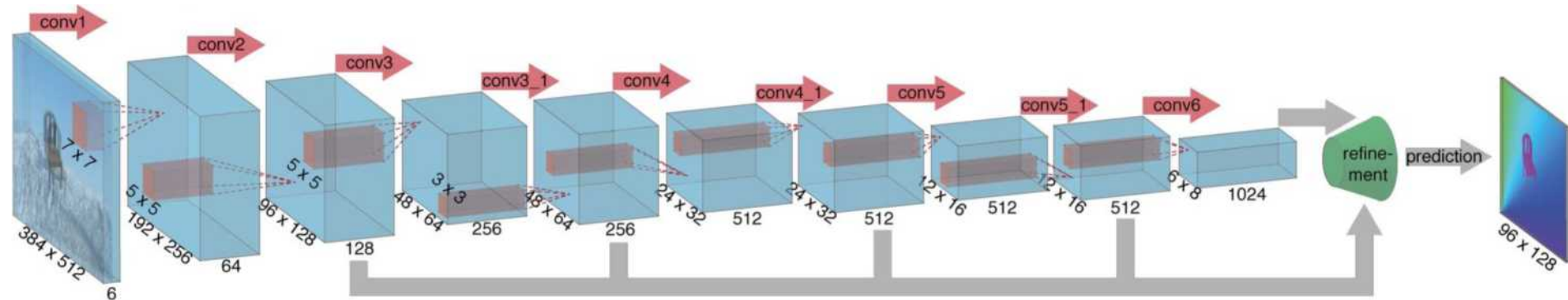
The network has to find if an object in the first image is at another location in the second images



FlowNet (encoder)

Option A: stack both input images together and feed them through a generic network.

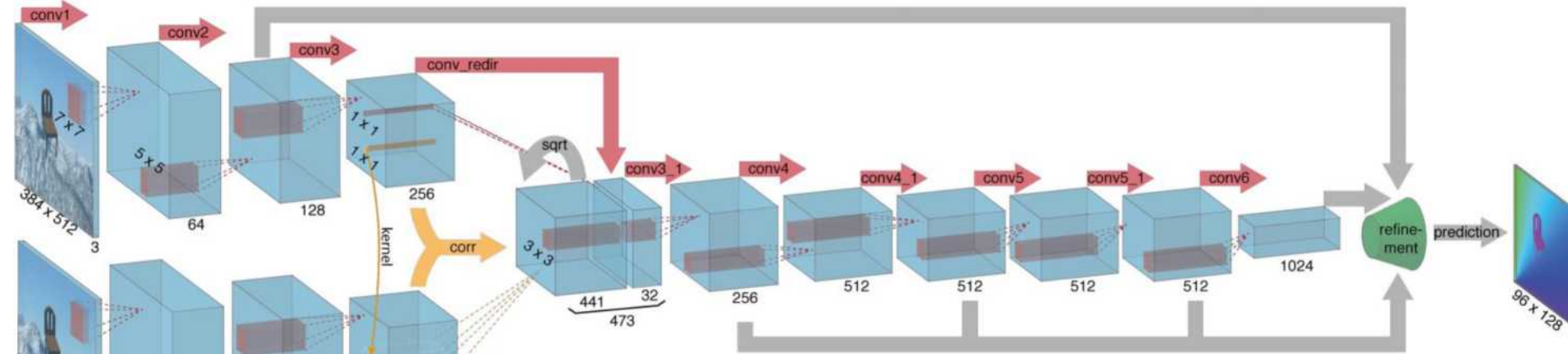
FlowNetSimple



FlowNet (encoder)

Option B: create two separate, yet identical processing streams for the two images and combine them at a later stage.

FlowNetCorr

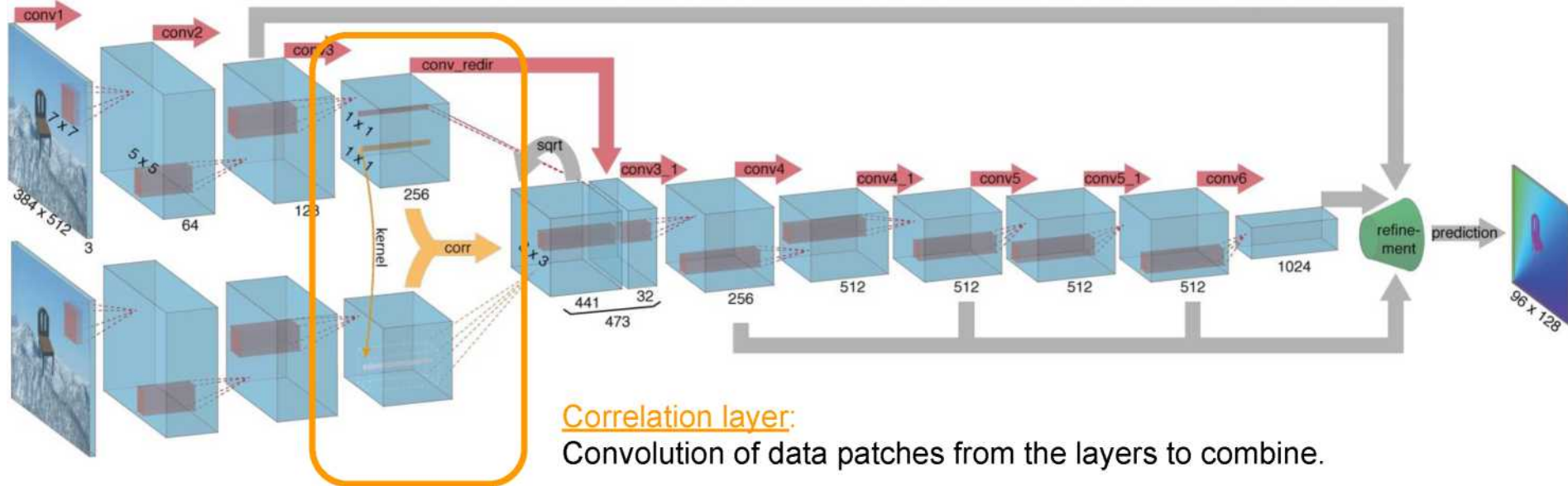


- At the later point features are combined further for flow estimation
- The network should be able to find the matching information before combination (added a combination layer with convolutional operation)
- The features are skip connected to the refinement layer

FlowNet (encoder)

Option B: create two separate, yet identical processing streams for the two images and combine them at a later stage.

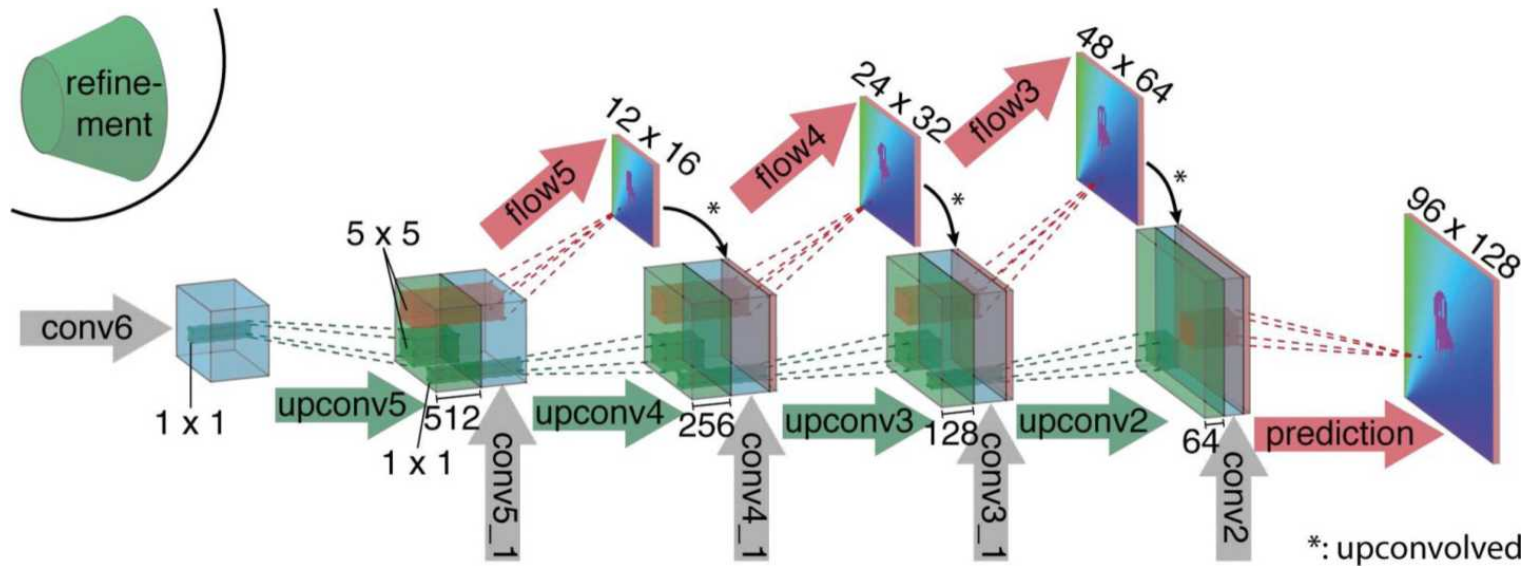
FlowNetCorr



FlowNet (decoder)

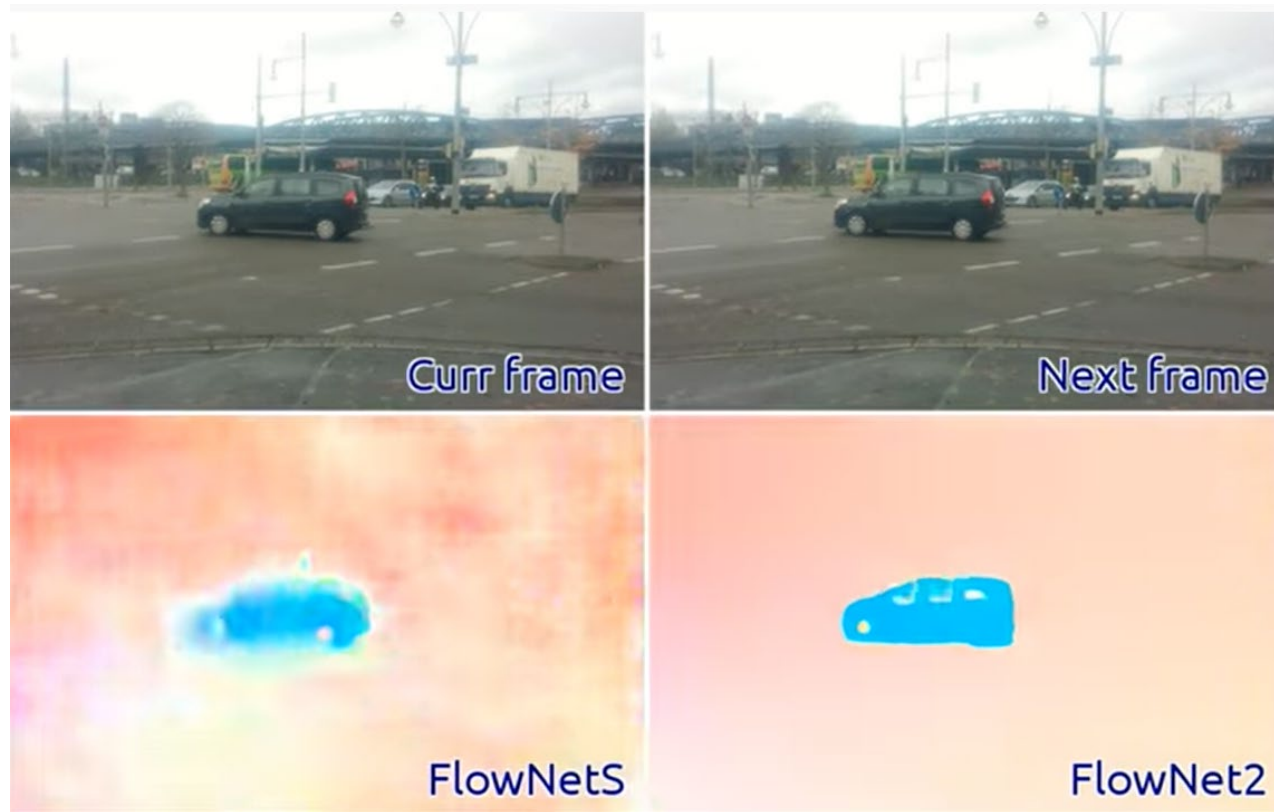
Upconvolutional layers: Unpooling features maps + convolution.

Upconvolutioned feature maps are concatenated with the corresponding map from the contractive part.

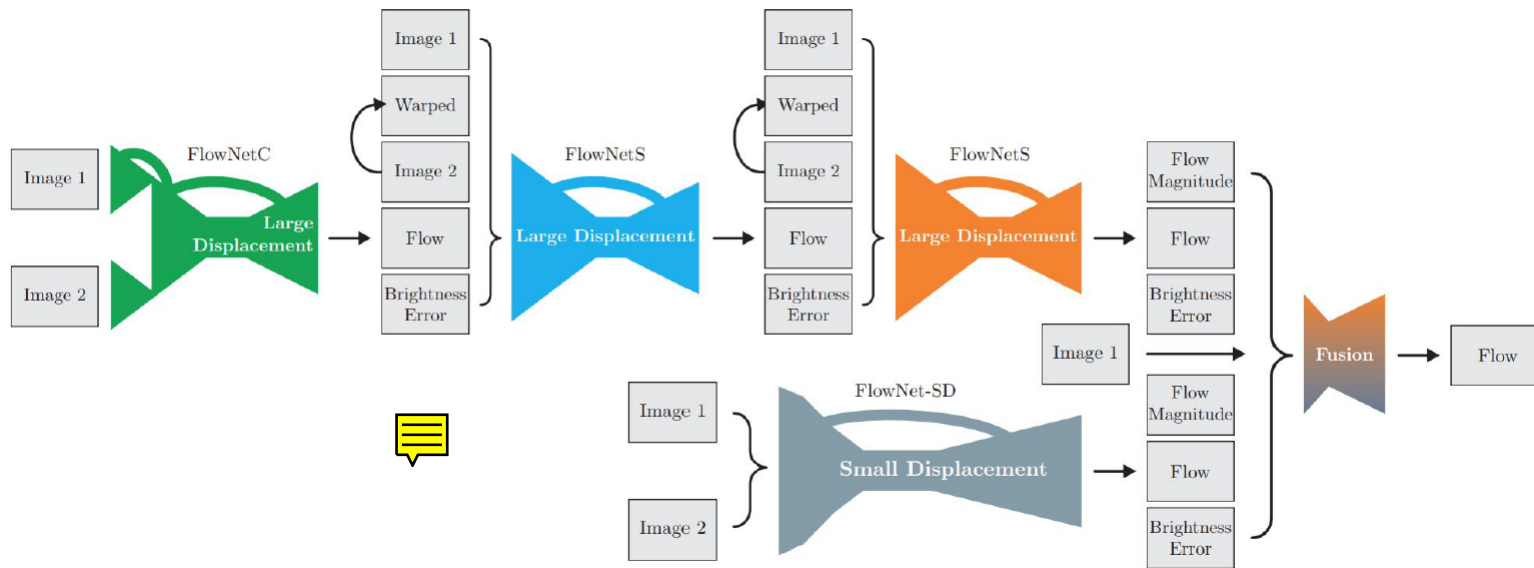


- Features map are scaled to the original resolution
- Upconvolutional layer (unpooling and convolution) doubling the resolution
- Concatenate the 'upconvolution' results with the features from the 'contractive' part of the network
- Bilinear upsampling (variational upsampling) in the last stage

FlowNet 2.0



Optical Flow: FlowNet 2.0

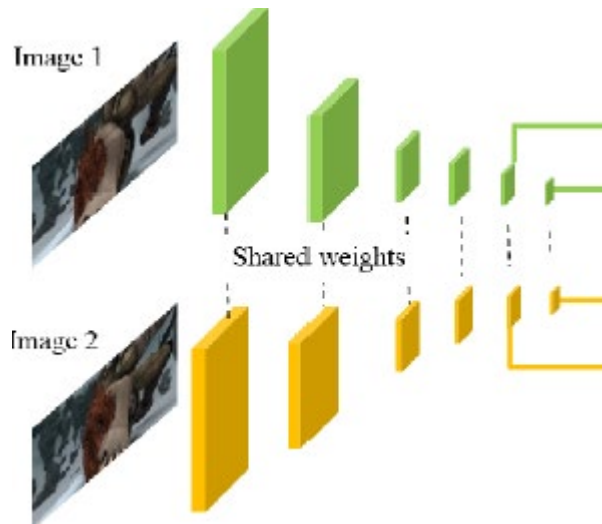


PWC-Net (Pyramid, Warping, Cost volume)

1. Feature extraction from input images with feature pyramid, i.e. convolutional layers
 - Reduction of spatial resolution
2. Optical flow estimation for every level of feature pyramid
 - Start with last convolutional layer, finish on input level
 - Warping and cost volume used in optical flow estimation

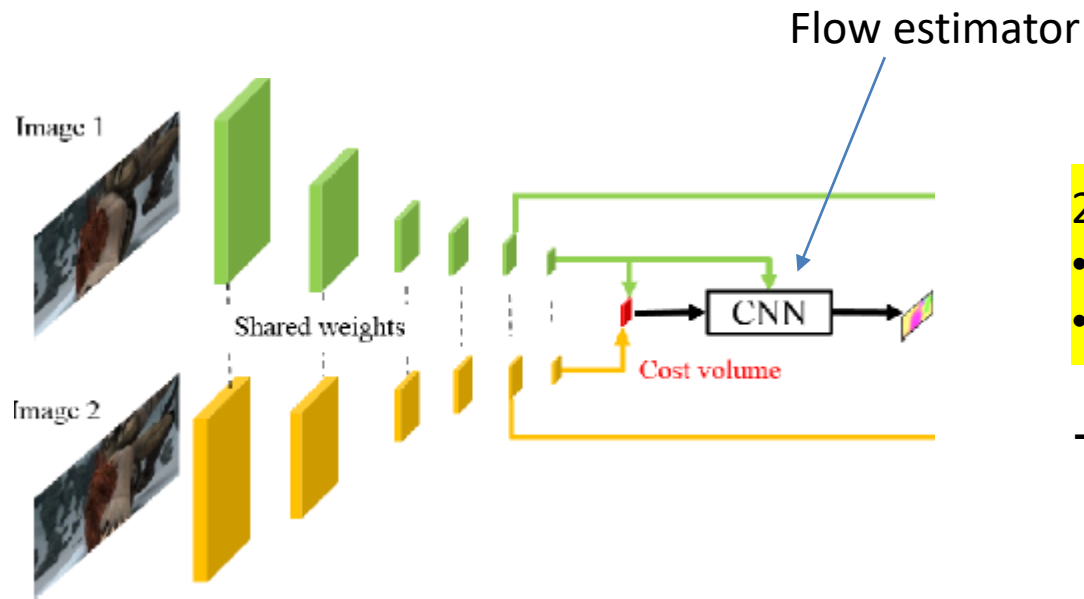


PWC-Net



1. Compute cost volume: find most similar pixel in features w.r.t other image
 - We can use normalized cross correlation to get the cost volume of a pixel
 - Invariance to color change, not in scale so pyramidal processing
 - Learning the features pyramid at lower scale we can learn global context to do the matching

PWC-Net

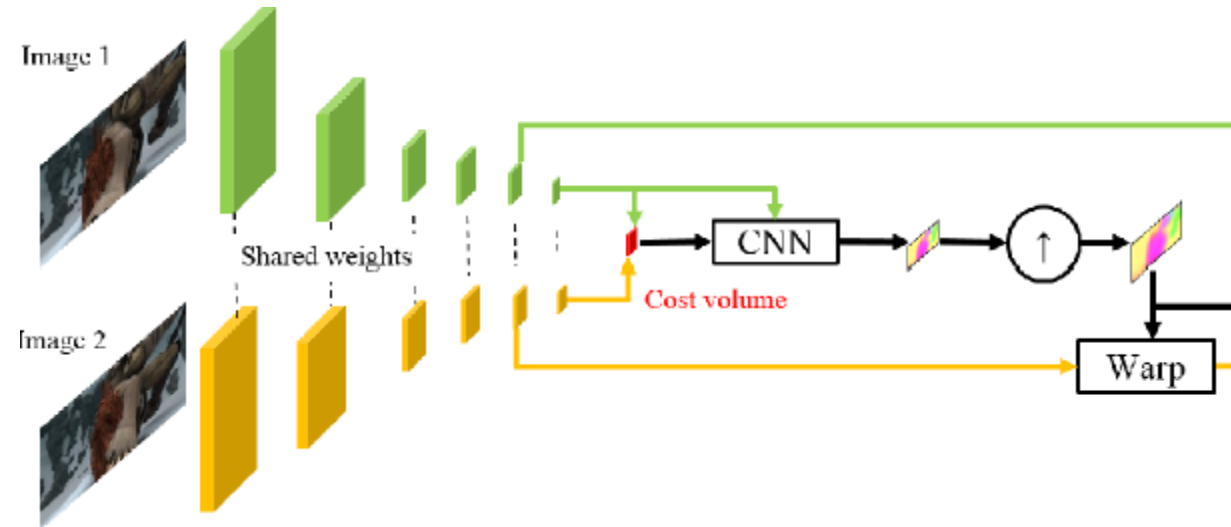


2. Optical flow estimation:

- cost volume
- features of the first image (image 1)

→ Output OF for lowest level

PWC-Net

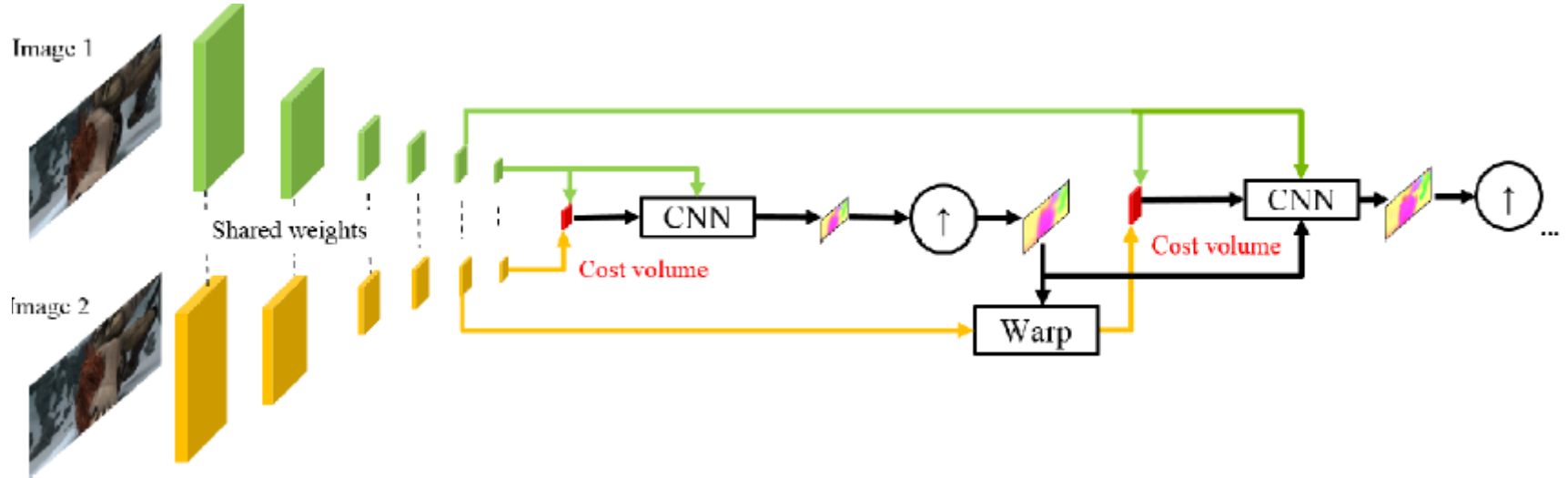


Why warping?

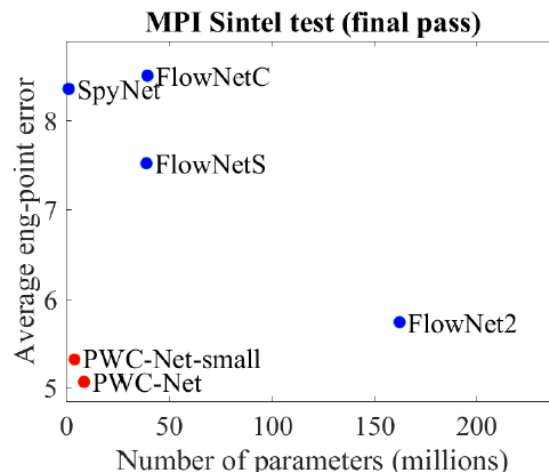
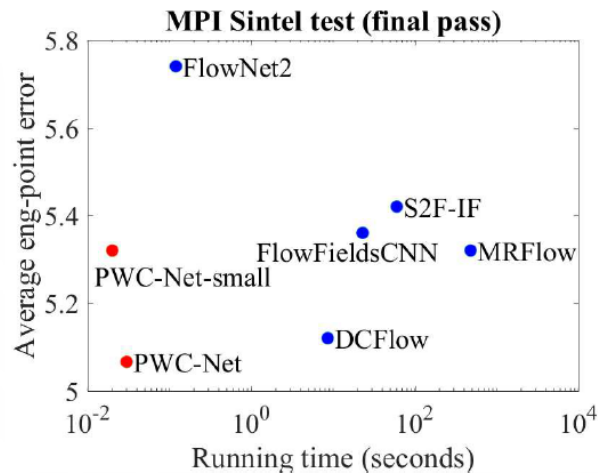
- Second image becomes more similar to first image
- Pixel displacement becomes smaller

1. Upsample OF to match spatial dimensions
2. Warp the features of the 2nd image towards the features of the 1st image (using the estimated flow field)

PWC-Net



PWC-Net results



- State-of-the-art accuracy with end-to-end training
- Inference fast enough for real-time application
- PWC-Net-small for mobile applications

Code publicly available: <https://github.com/NVlabs/PWC-Net>

Acknowledgement: some slides and material from Bernt Schiele, Mario Fritz, Michael Black, Bill Freeman, Fei-Fei, Justin Johnson, Serena Yeung, R. Szelisky, Fabio Galasso, Ioannis Gkioulekas, Noah Snavely, Abe Davis, Kris Kitani, Xavier Giró-i-Nieto