FlowNet: Learning Optical Flow with Convolutional Networks

Paper by : Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg , Philip Hausser, Caner Hazırbas, Vladimir Golkov

Presented BY Tushar Nimbhorkar and Alexander Lell

Outline

- Introduction
    - Introduction to the Optical Flow
    - Usage of Optical Flow
    - Motivation
    - Requirements for Optical Flow
    - Overview of the method in Paper
    - Related Work
- Method
    - Network architecture
    - FlowNetS
    - Refinement
    - FlowNetC
- Experiments & Results
    - Datasets
    - experiments
    - results
    - conclusion and follow-up research (FlowNet 2.0)

Introduction

- What is Optical Flow?

- The pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene.

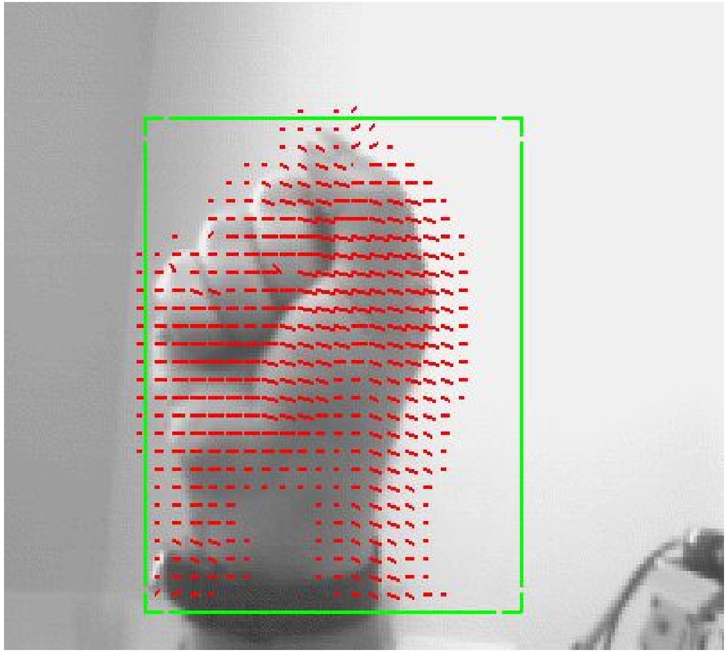
[Video result of this paper]

Where can it be used?

Object Tracking

Face tracking

Robotics



Motivation

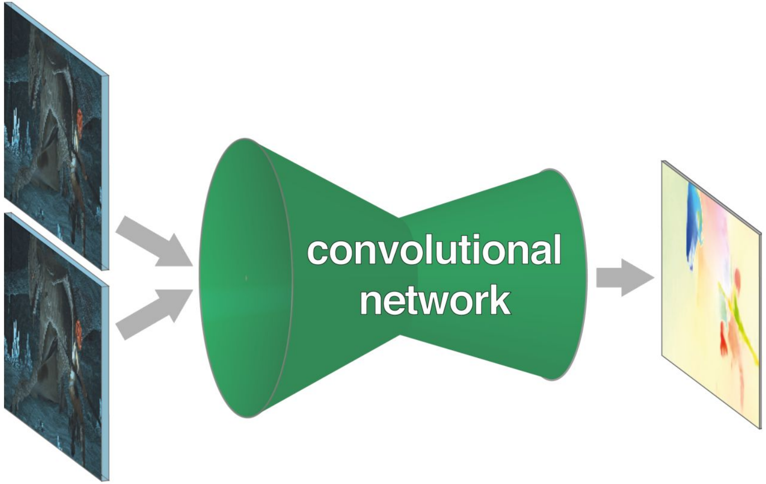
- Convolutional neural networks in computer vision problems.
- Main Idea/Method: Train CNN end-to-end to learn predicting optical flow field for a pair of images.

Requirements for Optical Flow

- Estimation needs per-pixel localization.
- Also need to find correspondences between the pair of images.

- It will involve not only learning image feature representations, but also learning to match them at different locations in the two images.

Overview of the Network



- Not sure whether this task could be solved with a standard CNN architecture
- Also developed architecture with a correlation layer that explicitly provides matching capabilities.

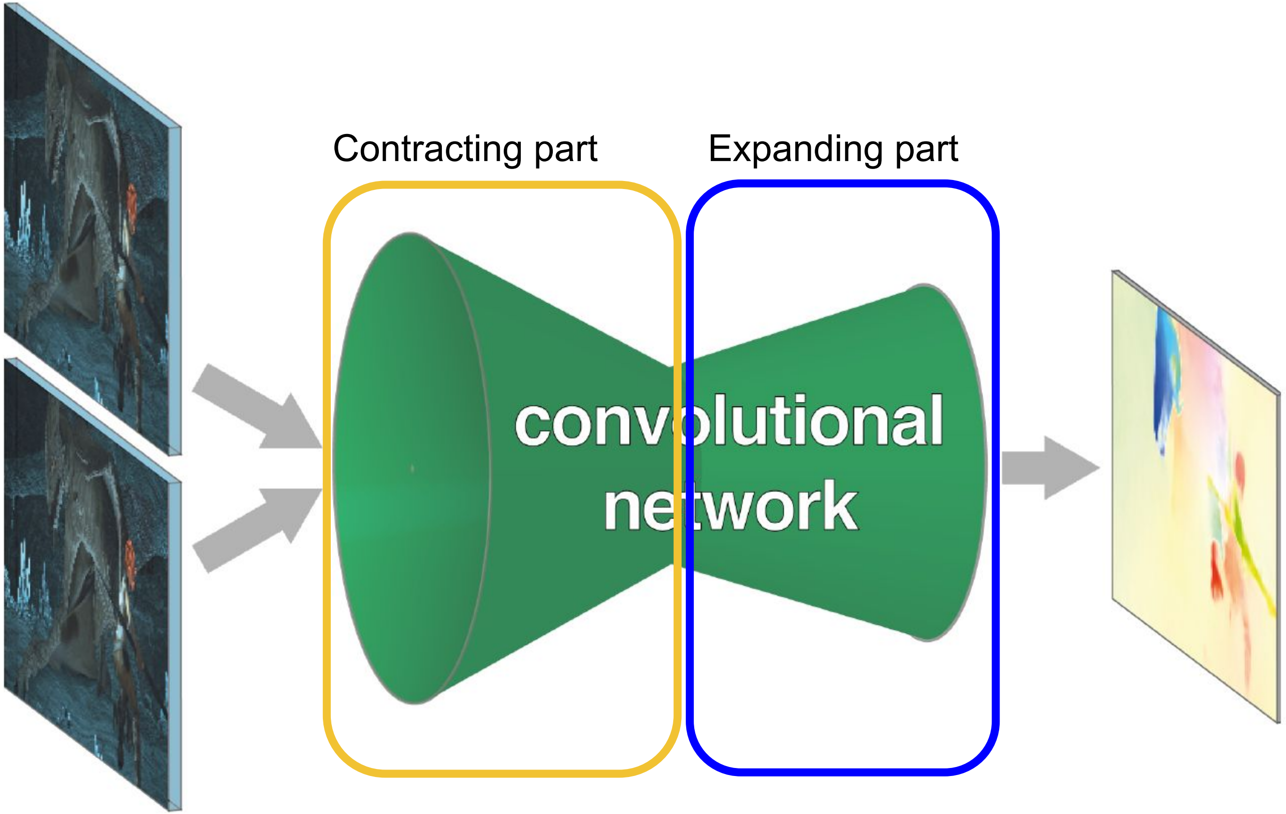
Related work

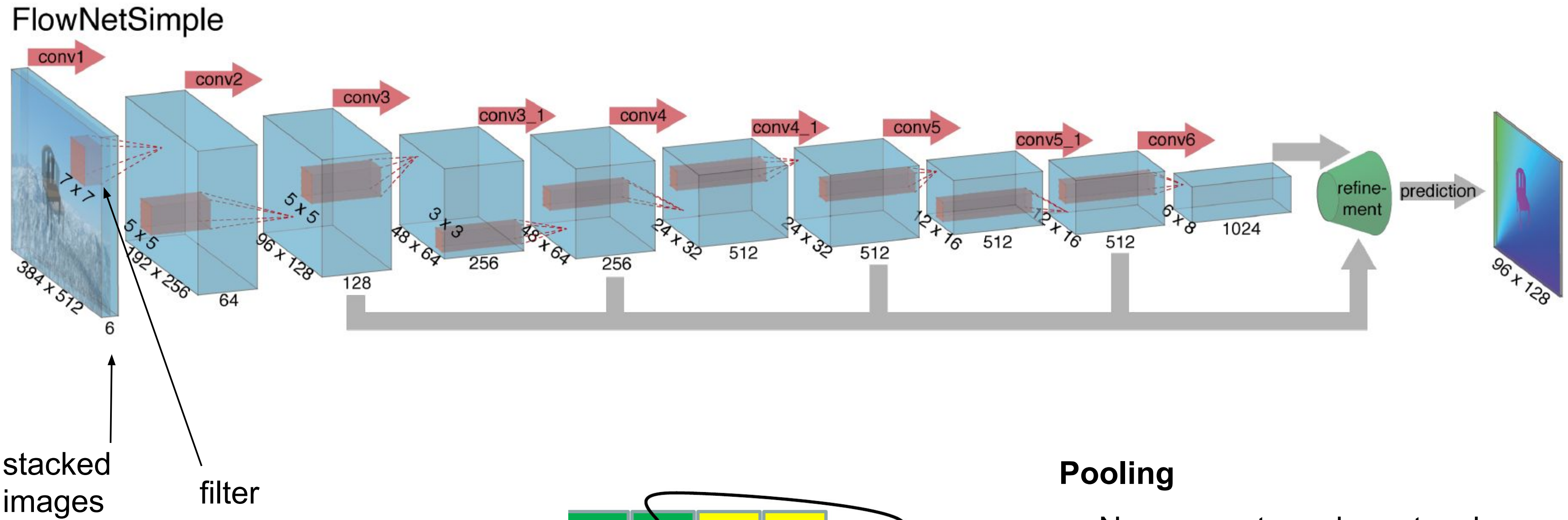
- Horn and Schunck(1981)
- Lucas-Kanade(1981)
- DeepMatching and DeepFlow (2013)
- EpicFlow (2015)
- Sun et al. (2008)

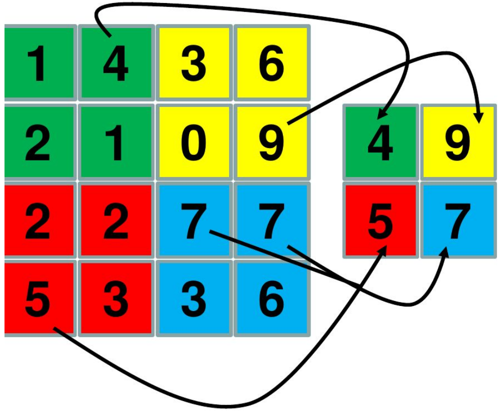
Continued.

- No direct work of predicting optical flow with CNNs.
- Why?

Network Architecture


Contracting part
Expanding part
convolutional
network

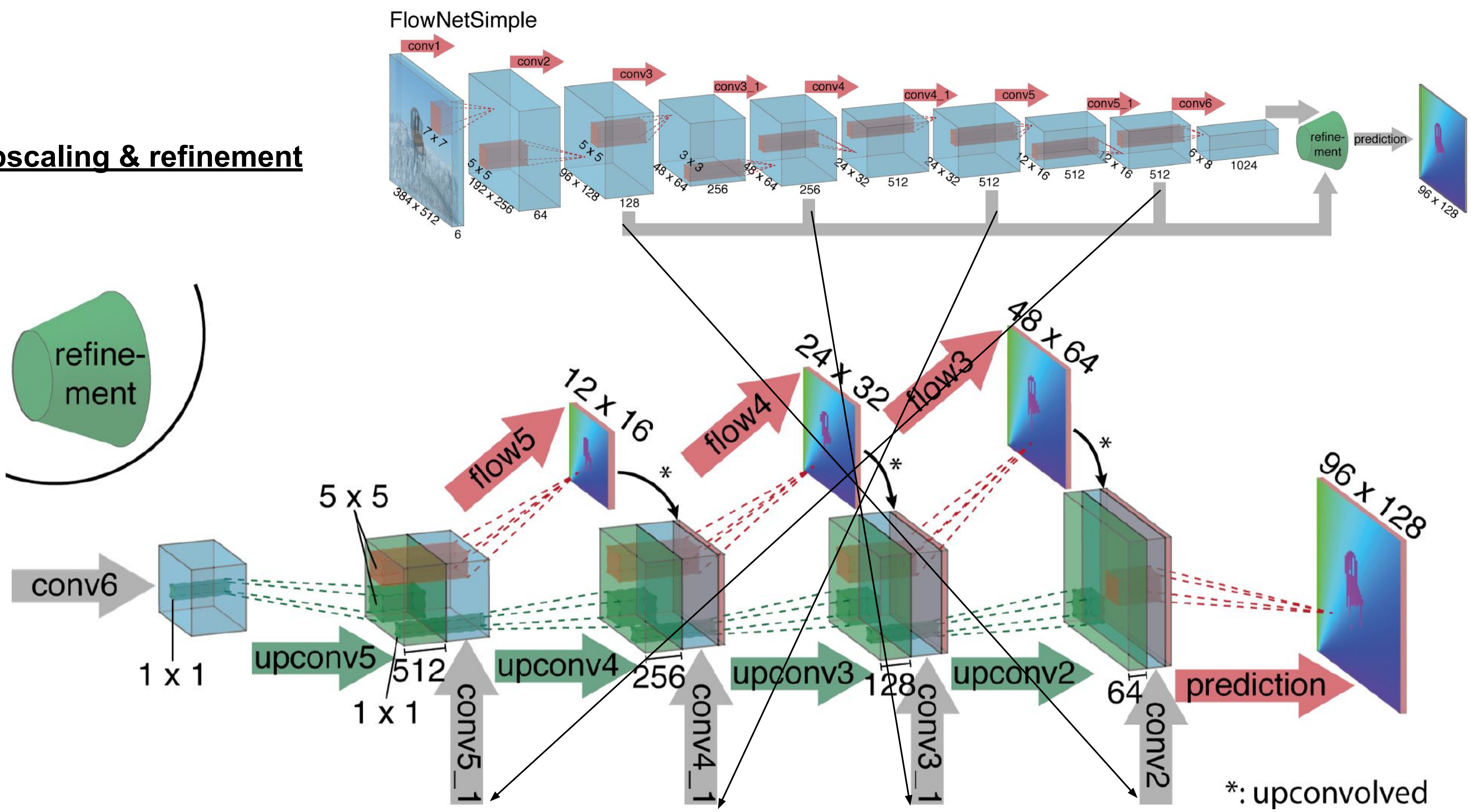


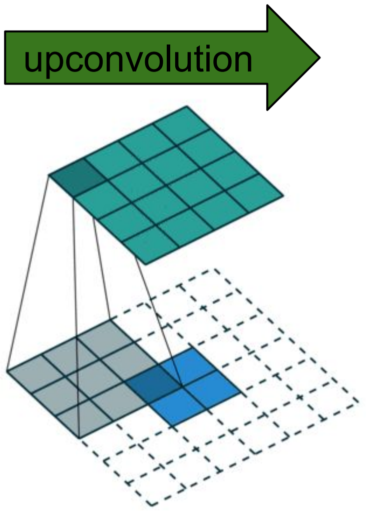


Pooling

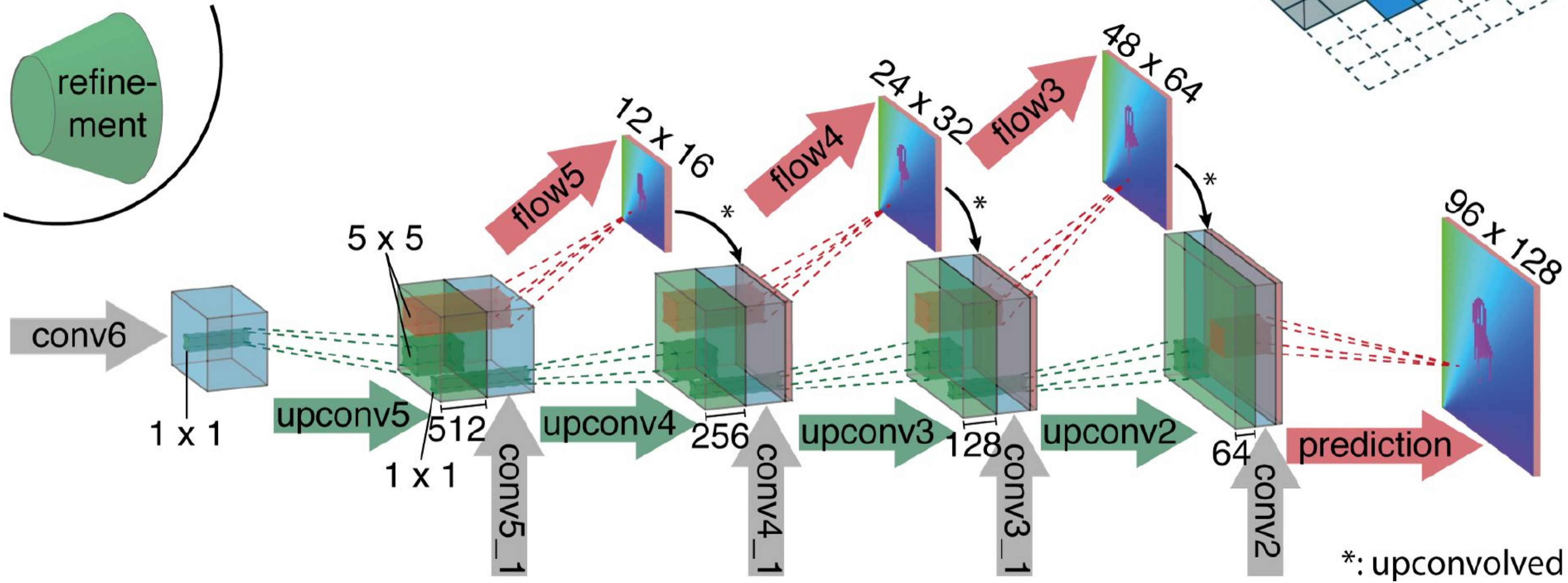
- Necessary to make network training computationally feasible
- Problem: reduces resolution
- Solution: refine coarse feature maps

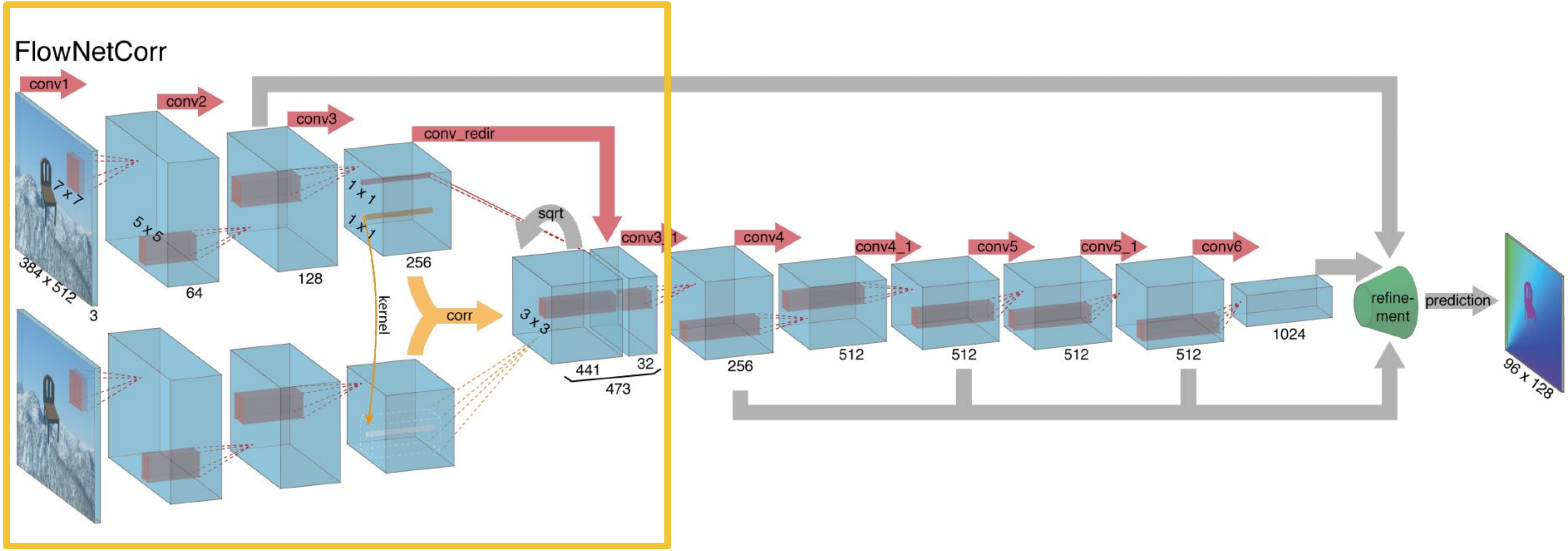
Upscaling & refinement

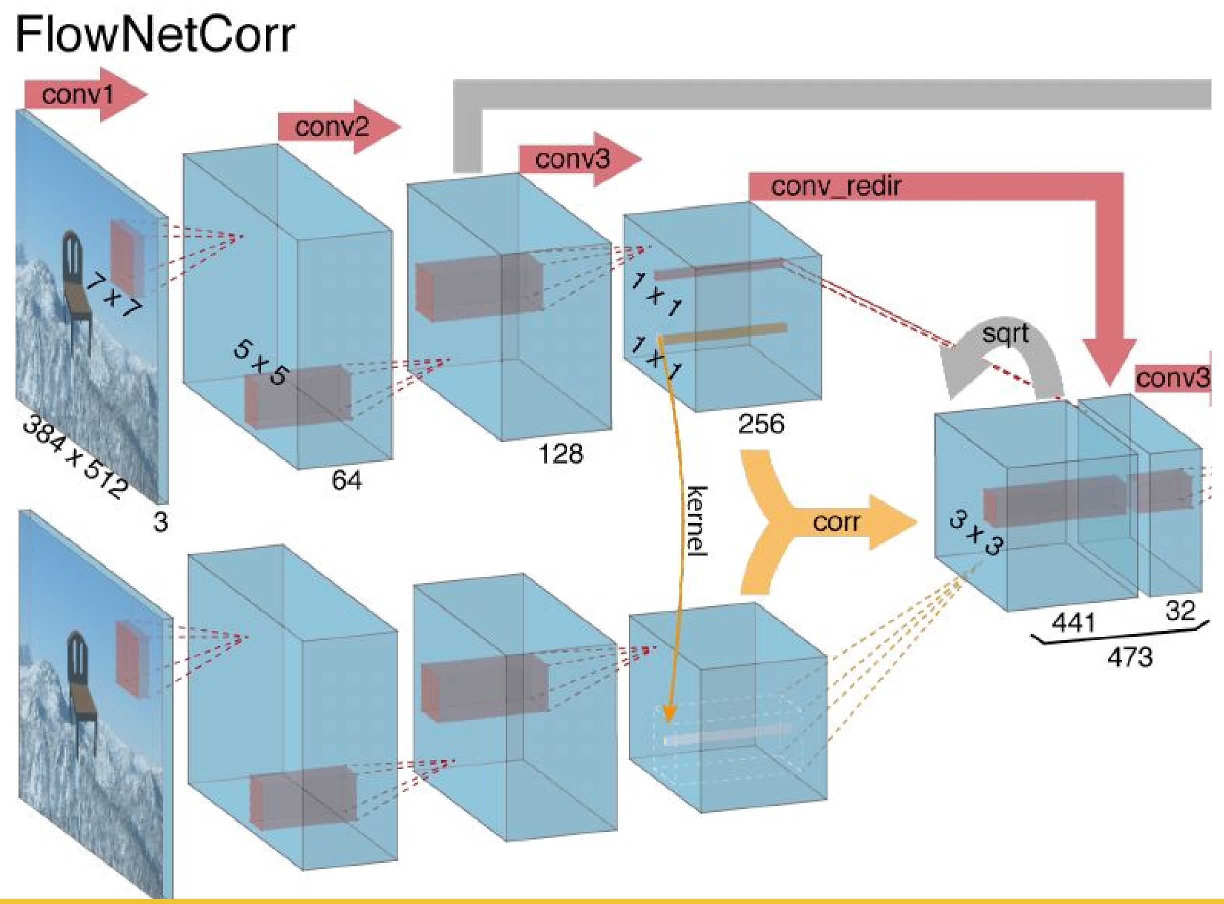




Upscaling & refinement



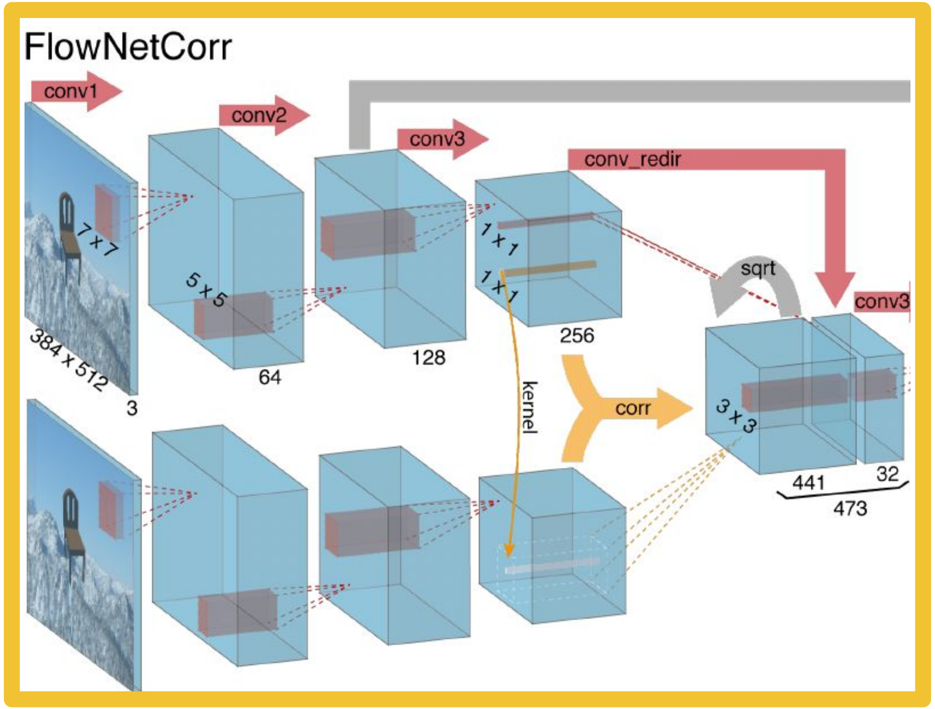

FlowNetCorr
conv1
conv2
conv3
conv_redir
7 x 7
5 x 5
1 x 1
1 x 1
384 x 512
3
64
128
256
kernel
corr
sqrt
3 x 3
441
32
473
conv3_1
conv4
conv4_1
conv5
conv5_1
conv6
256
512
512
512
512
1024
refine-
ment
prediction
96 x 128



FlowNetCorr
conv1
conv2
conv3
conv_redir
7 x 7
5 x 5
1 x 1
1 x 1
384 x 512
3
64
128
256
kernel
corr
sqrt
conv3
3 x 3
441
32
473

Correlation

$$c(\mathbf{x}_1, \mathbf{x}_2) = \sum_{\mathbf{o} \in [-k,k] \times [-k,k]} \langle \mathbf{f}_1(\mathbf{x}_1 + \mathbf{o}), \mathbf{f}_2(\mathbf{x}_2 + \mathbf{o}) \rangle$$



Dataset

	Frame pairs	Frames with ground truth	Ground truth density per frame
Middlebury	72	8	100%
KITTI	194	194	∽50%
Sintel	1,041	1,041	100%

Dataset

Experiment

Convolution layer : 9

Stride : 2 (Only in six of them)

Nonlinearity : ReLu (After each layer)

Filter sizes: Decreases as we go deeper in network.(7X7 to 3X3)

Training loss: Endpoint error (EPE)

Optimization Method: Adam

Results

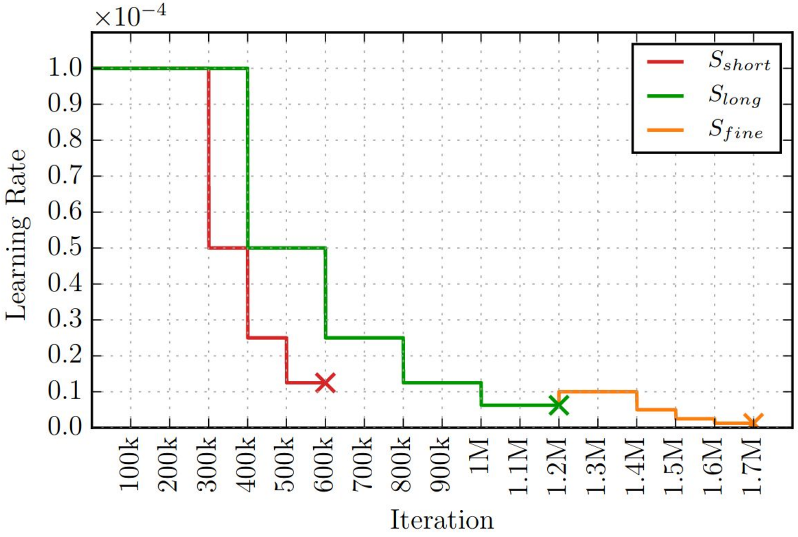
Method	Sintel Clean		Sintel Final		KITTI		Middlebury train		Middlebury test		Chairs	Time (sec)	
	train	test	train	test	train	test	AEE	AAE	AEE	AAE	test	CPU	GPU
EpicFlow [30]	2.27	4.12	3.57	6.29	3.47	3.8	0.31	3.24	0.39	3.55	2.94	16	-
DeepFlow [35]	3.19	5.38	4.40	7.21	4.58	5.8	0.21	3.04	0.42	4.22	3.53	17	-
EPPM [3]	-	6.49	-	8.38	-	9.2	-	-	0.33	3.36	-	-	0.2
LDOF [6]	4.19	7.56	6.28	9.12	13.73	12.4	0.45	4.97	0.56	4.55	3.47	65	2.5
FlowNetS	4.50	7.42	5.45	8.43	8.26	-	1.09	13.28	-	-	2.71	-	0.08
FlowNetS+v	3.66	6.45	4.76	7.67	6.50	-	0.33	3.87	-	-	2.86	-	1.05
FlowNetS+ft	(3.66)	6.96	(4.44)	7.76	7.52	9.1	0.98	15.20	-	-	3.04	-	0.08
FlowNetS+ft+v	(2.97)	6.16	(4.07)	7.22	6.07	7.6	0.32	3.84	0.47	4.58	3.03	-	1.05
FlowNetC	4.31	7.28	5.87	8.81	9.35	-	1.15	15.64	-	-	2.19	-	0.15
FlowNetC+v	3.57	6.27	5.25	8.01	7.45	-	0.34	3.92	-	-	2.61	-	1.12
FlowNetC+ft	(3.78)	6.85	(5.28)	8.51	8.79	-	0.93	12.33	-	-	2.27	-	0.15
FlowNetC+ft+v	(3.20)	6.08	(4.83)	7.88	7.31	-	0.33	3.81	0.50	4.52	2.67	-	1.12

Conclusion

- It is possible to train Network to directly predict OpticalFlow
- Even if training set in not real.
- On synthetic Test set : CNNs Outperforms state-of-the-art Methods.

Can we do better? Yes, with [FlowNet 2.0]

- Realistic training data and improved training schedule
    - More iterations during training
    - More realistic training data only presented during fine tuning
    - First, the network learns basic features, then ones more refined



If at first you don't succeed.
Try, try and try again (with more networks).