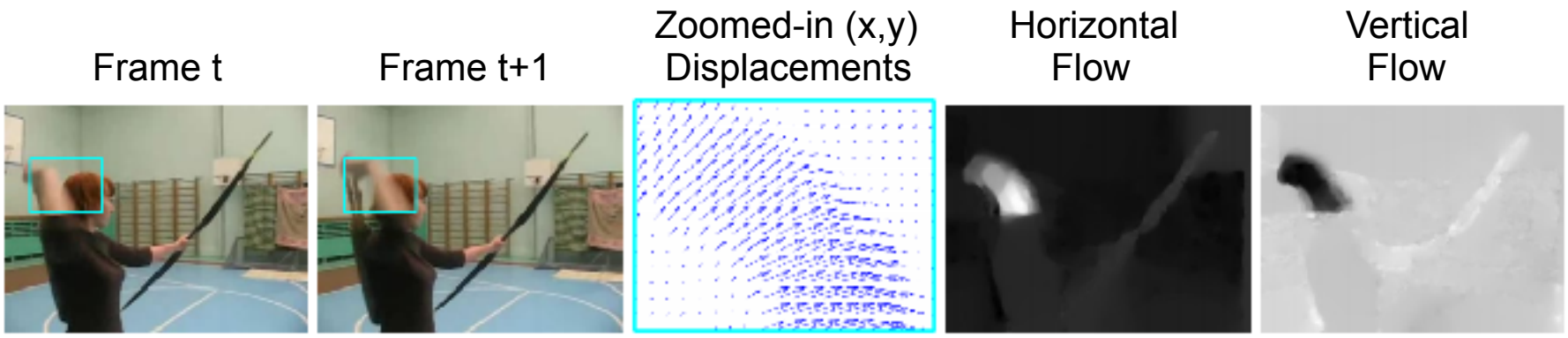
FlowNet: Learning Optical Flow with Convolutional Networks

ICCV 2015

Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, Thomas Brox

Problem Overview

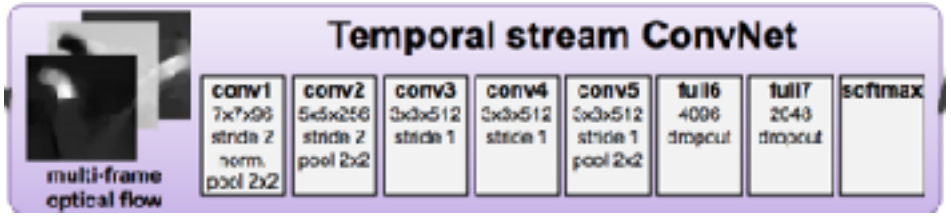
- Given two adjacent frames we want to predict an optical flow field for those two frames.



Motivation

- Optical flow can be useful for many different applications.

Object detection & tracking



Action classification

Video generation

Prior Work

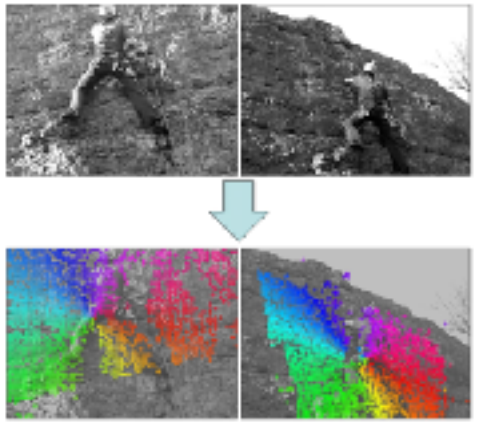
- Optimization based strategies to obtain optical flow.
- CNN-based approaches that don't involve any end-to-end learning.

$$E(v) = \int_{\Omega} \left(\nabla I^{\top} v + I_t\right)^2 + \lambda \left(|\nabla v_1|^2 + |\nabla v_2|^2\right) d^2x \,.$$

Determining optical flow [Artificial Intelligence 1981]

$$E_{Data}(u,v) = \int_{\Omega} \Psi \left(|I(\mathbf{x}+\mathbf{w}) - I(\mathbf{x})|^2 + \gamma |\nabla I(\mathbf{x}+\mathbf{w}) - \nabla I(\mathbf{x})|^2\right) \mathbf{dx}$$

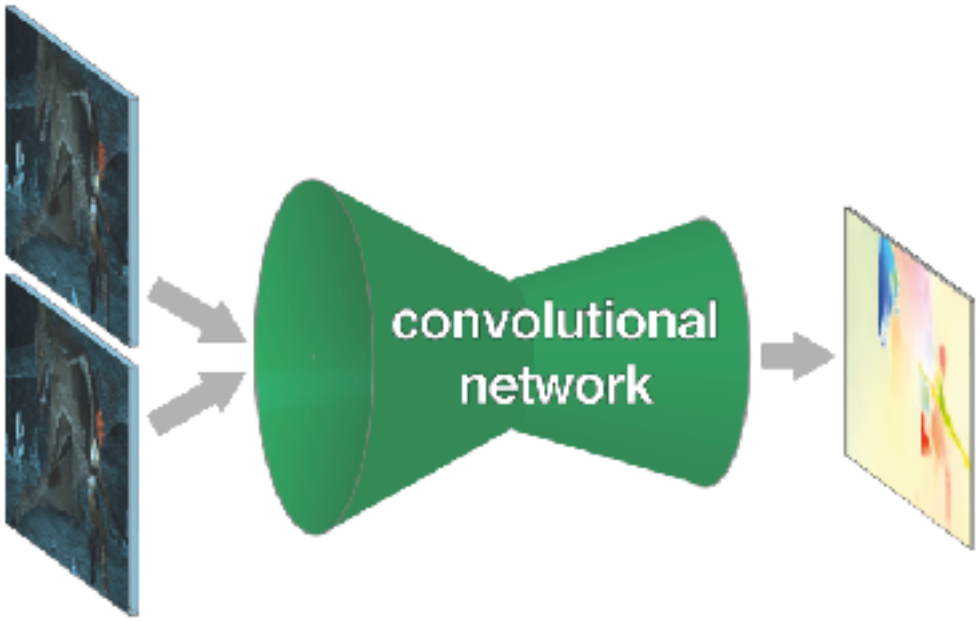
High accuracy optical flow estimation based on a theory for warping [ECCV 2004]



DeepFlow: Large displacement optical flow with deep matching [ICCV 2013]

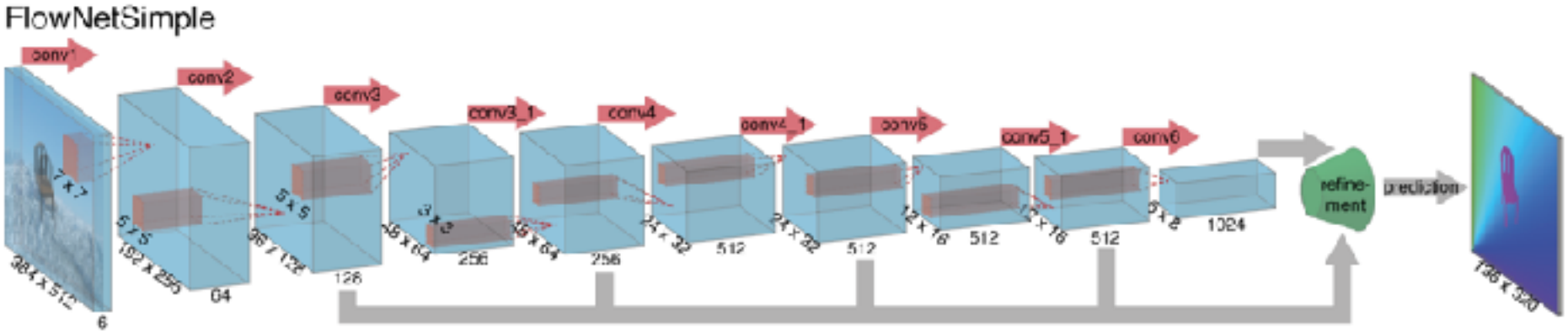
Challenges

- Can CNNs learn to find correspondences between pixels in two images?
- The existing optical flow datasets are too small for effective CNN training.



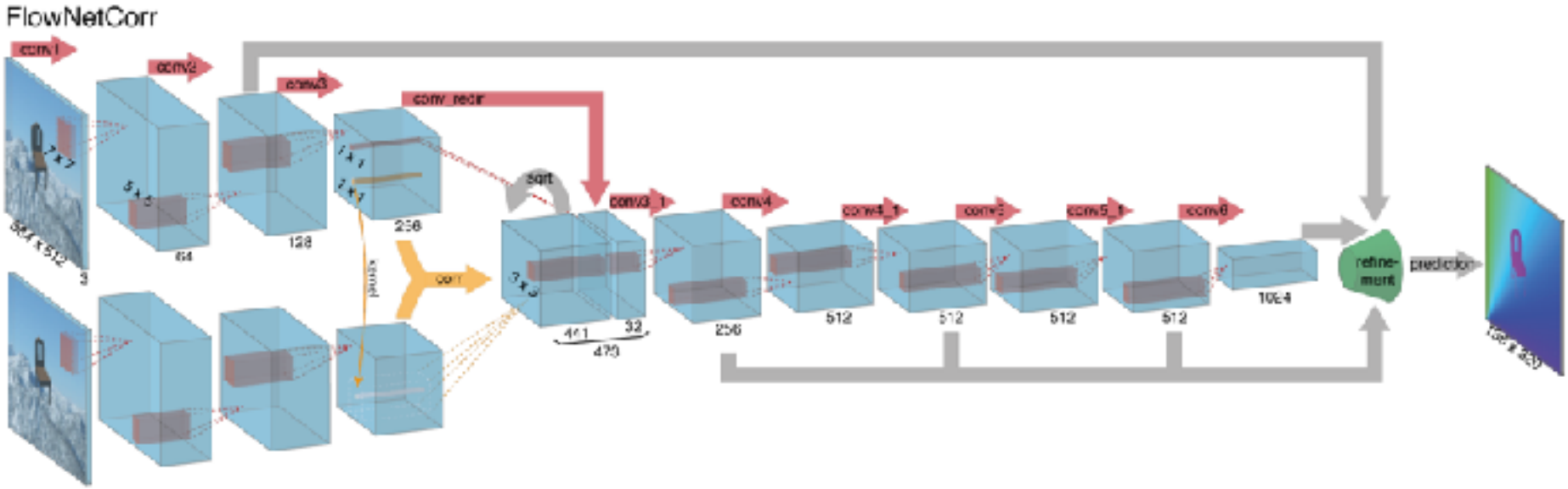
FlowNetS (Simple Network)

- Stack both input images together and feed them through a generic CNN network.
- Can a model with such a generic architecture learn to solve an optical flow task?



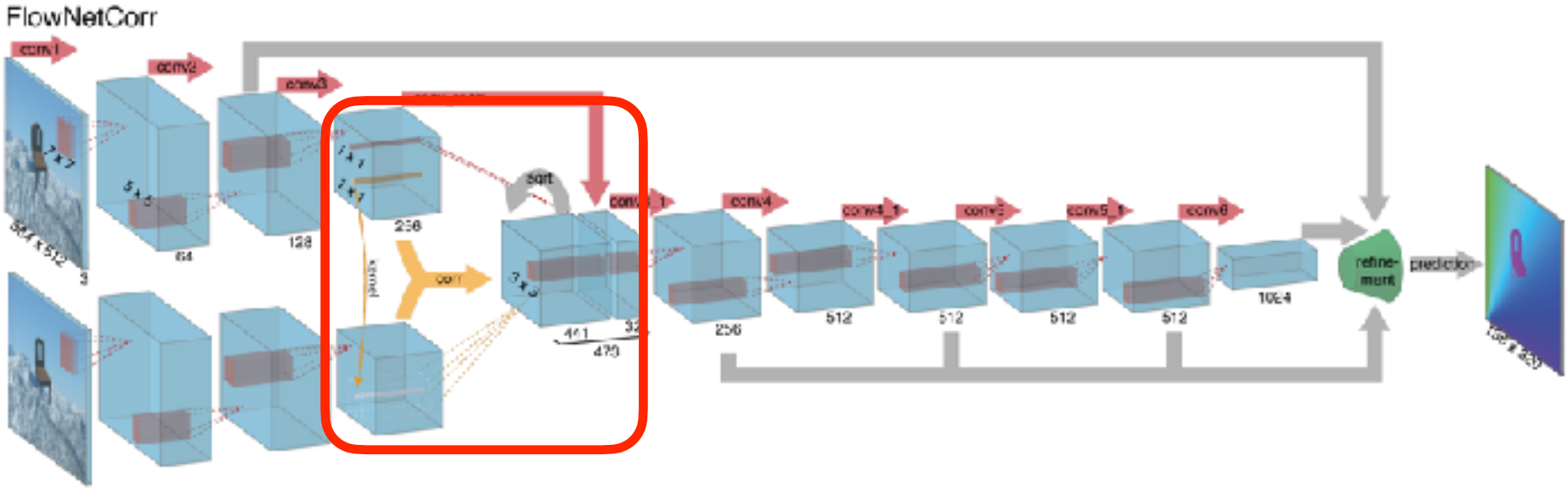
FlowNetC (Correlation Network)

- Create two separate, yet identical processing streams for the two images.
- Combine them at a later stage via a correlation layer.



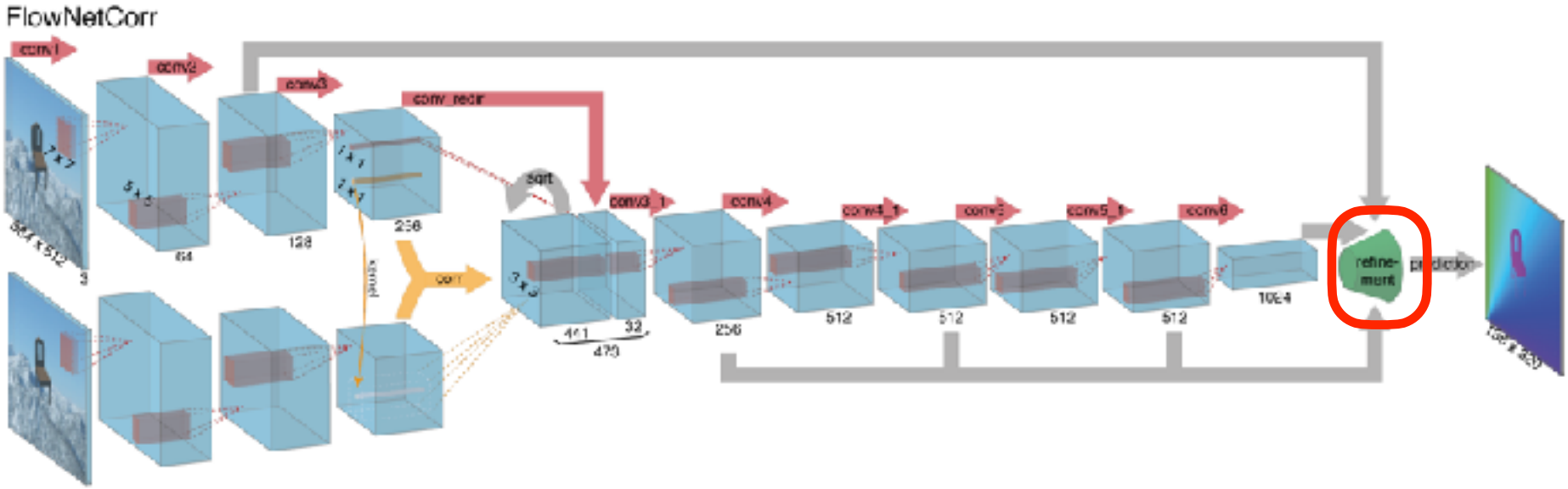
FlowNetC (Correlation Network)

- Create two separate, yet identical processing streams for the two images.
- Combine them at a later stage via a correlation layer.



FlowNetC (Correlation Network)

- Create two separate, yet identical processing streams for the two images.
- Combine them at a later stage via a correlation layer.



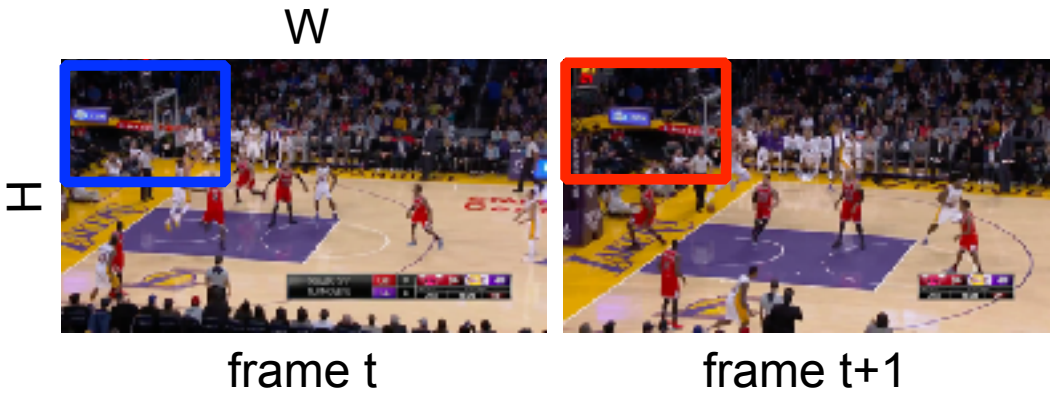
Correlation Layer

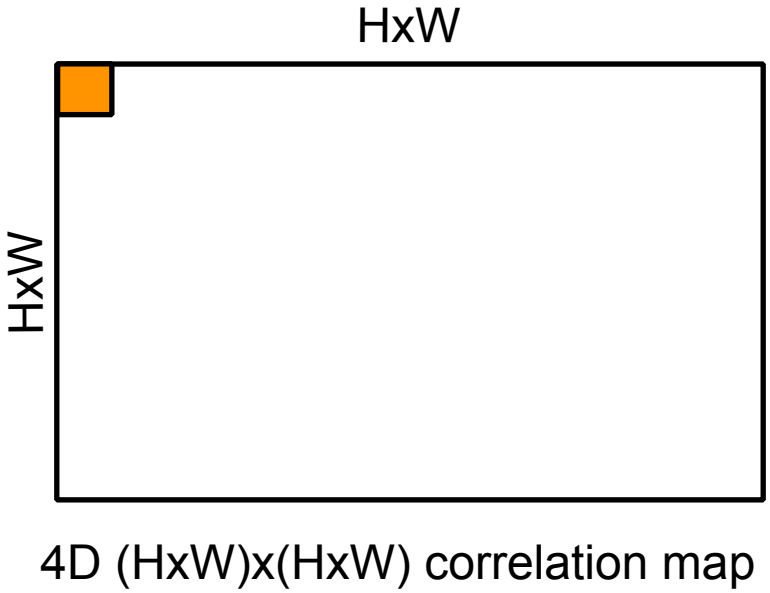
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

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

Correlation Layer

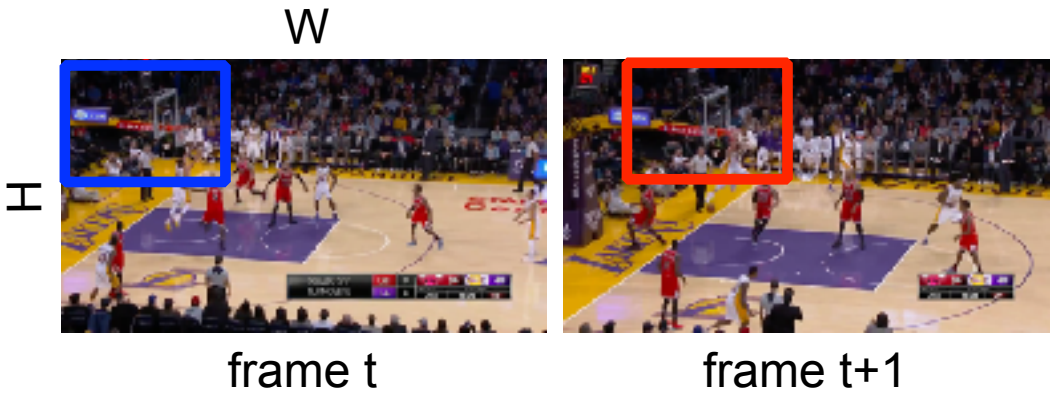
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

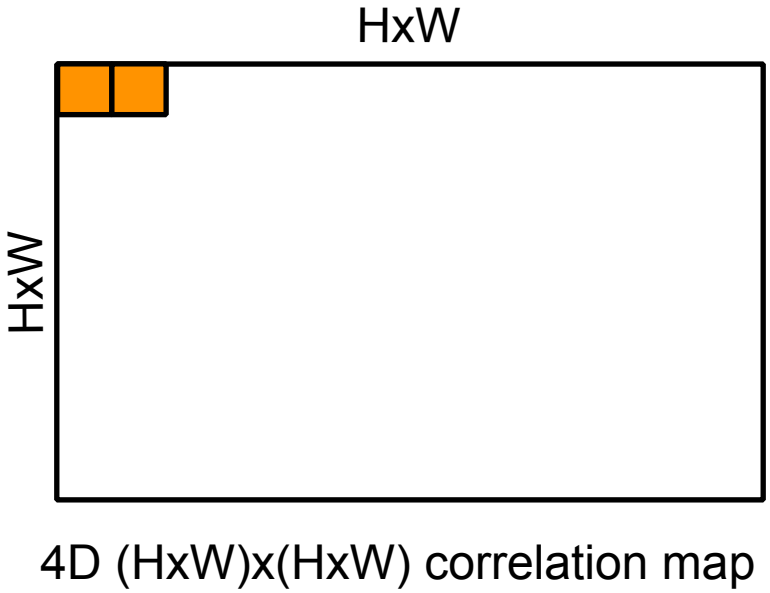




Correlation Layer

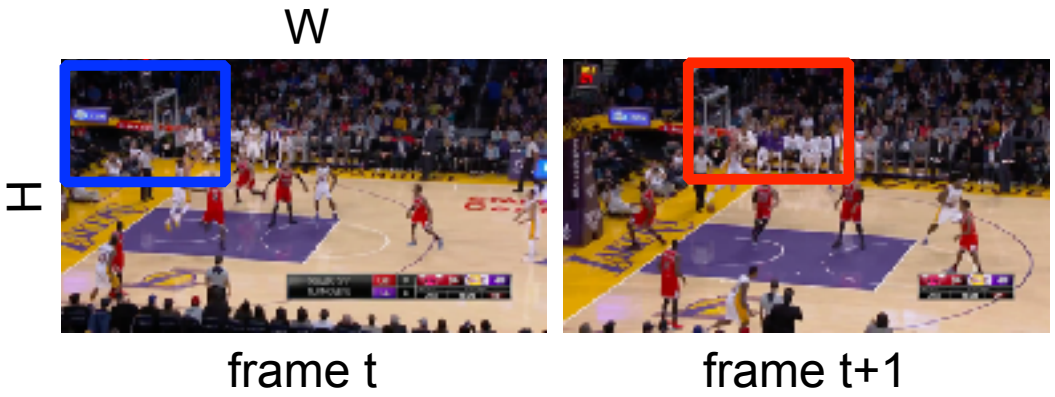
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

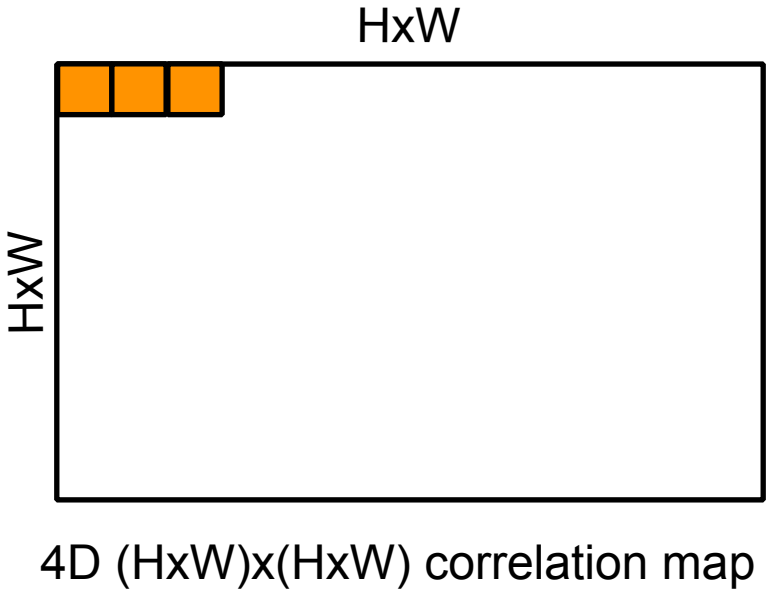




Correlation Layer

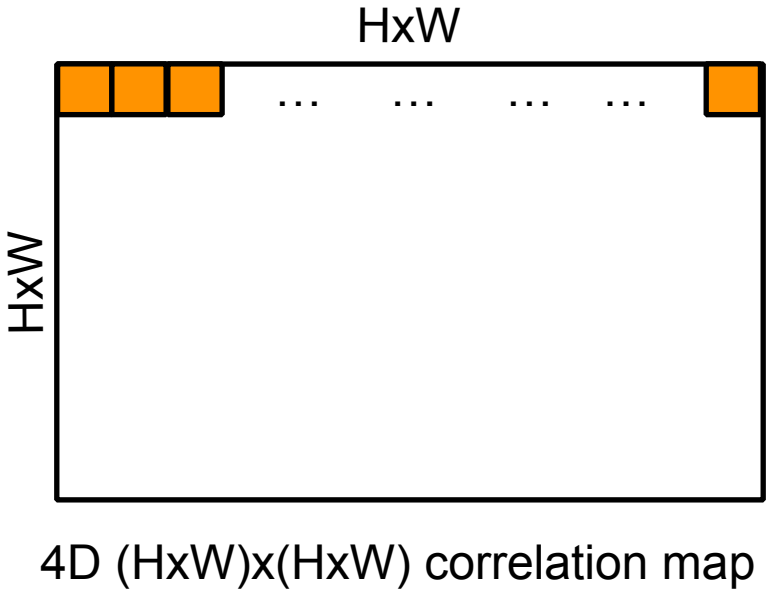
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.





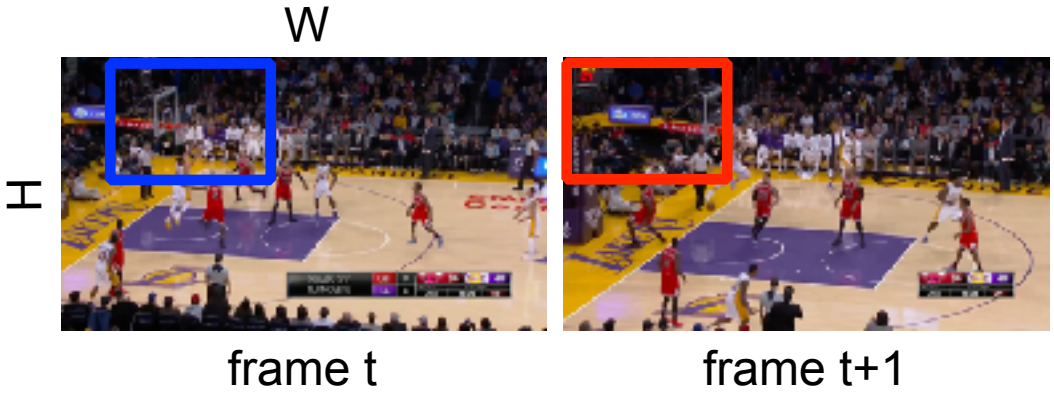
Correlation Layer

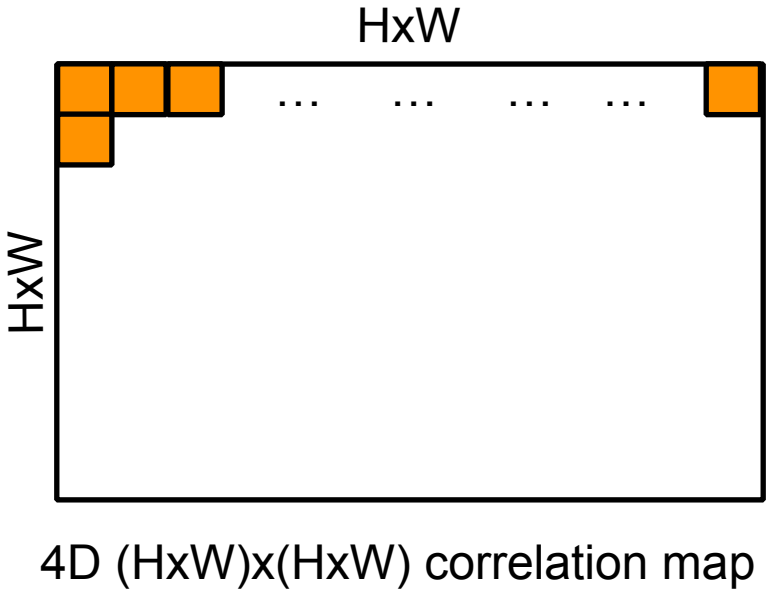
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.



Correlation Layer

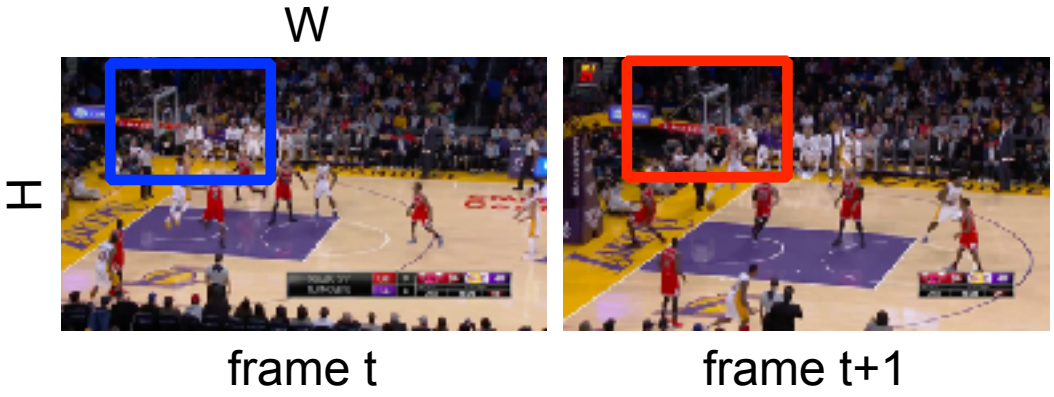
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

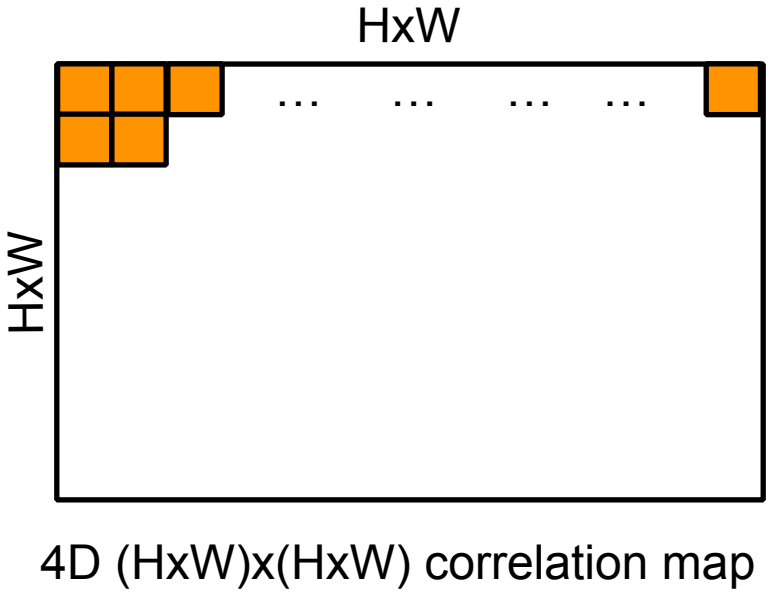




Correlation Layer

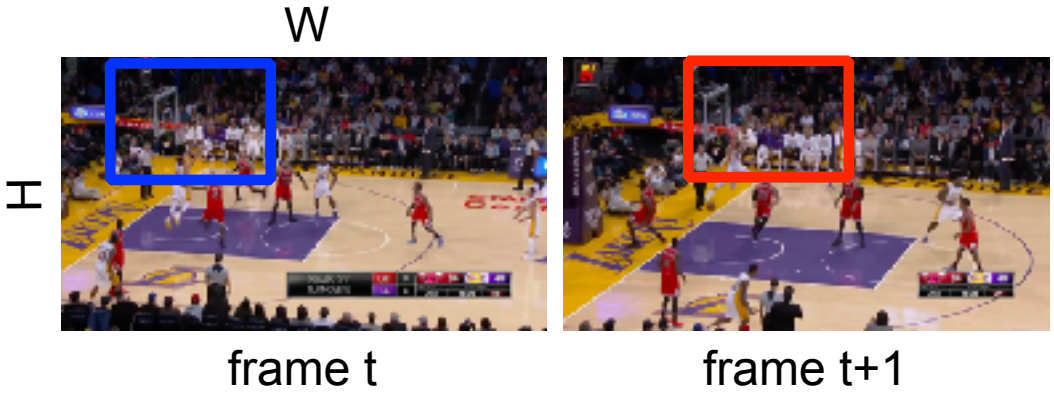
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

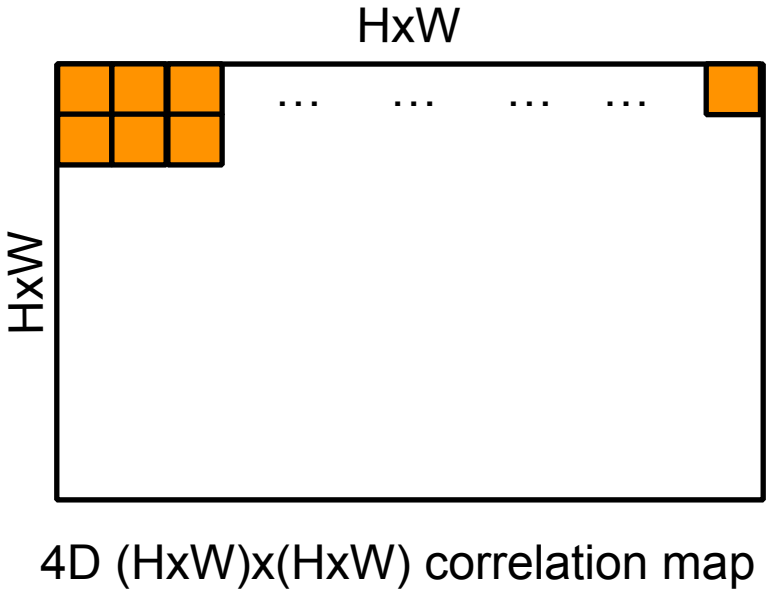




Correlation Layer

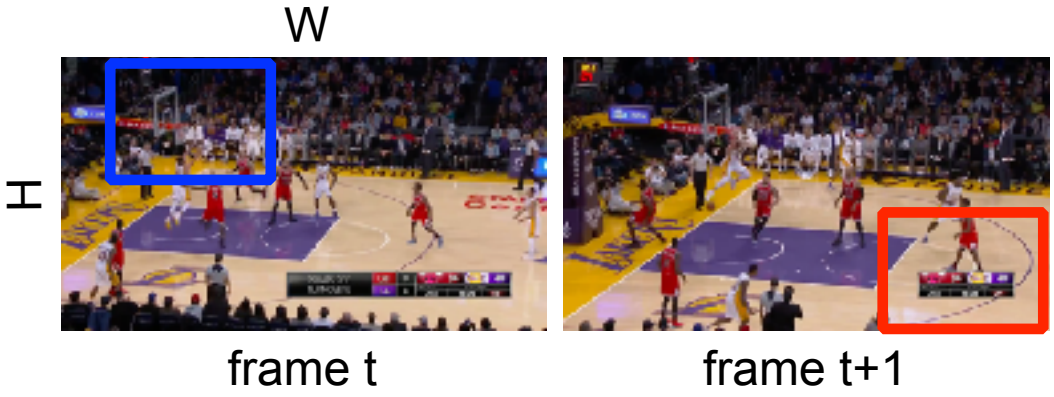
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

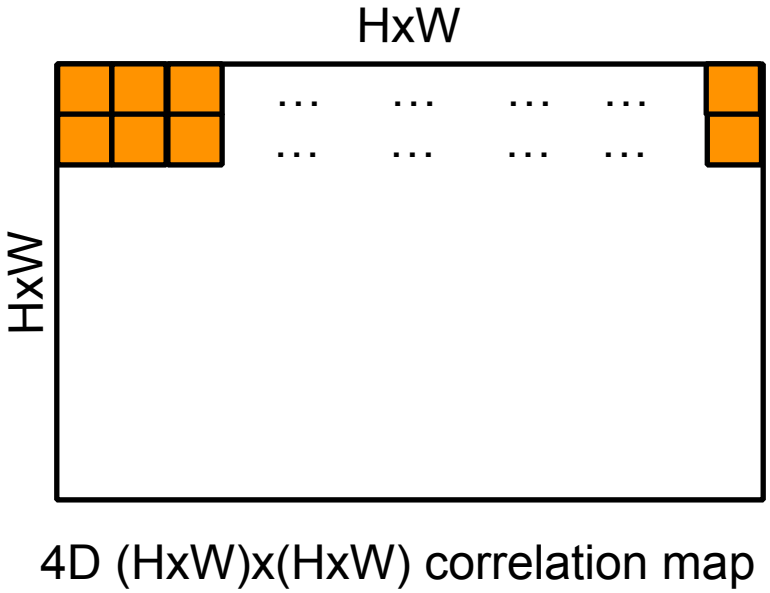




Correlation Layer

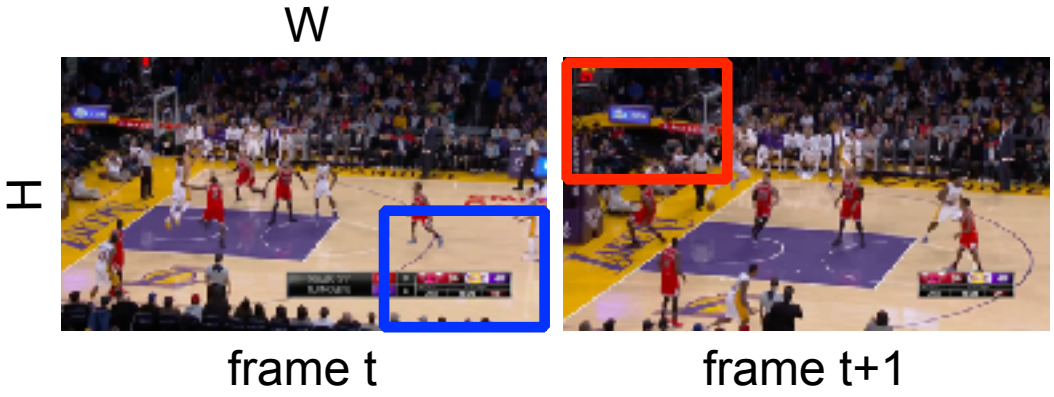
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

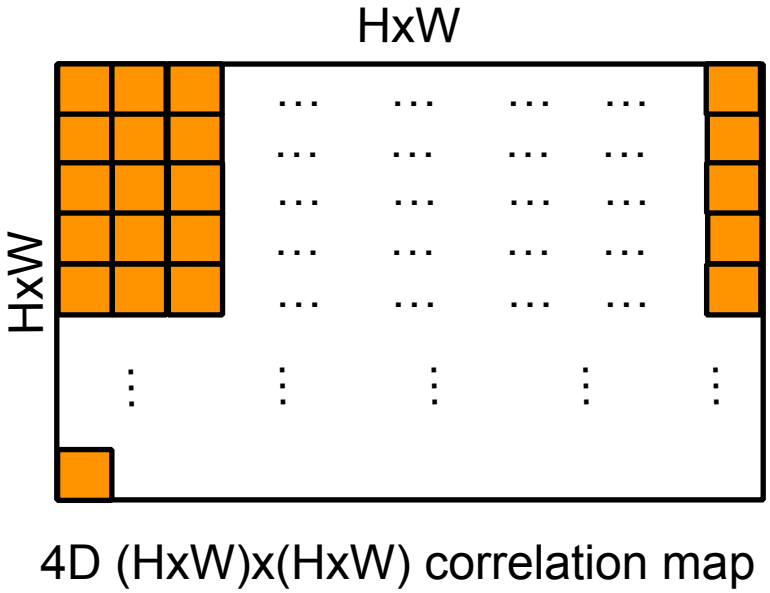




Correlation Layer

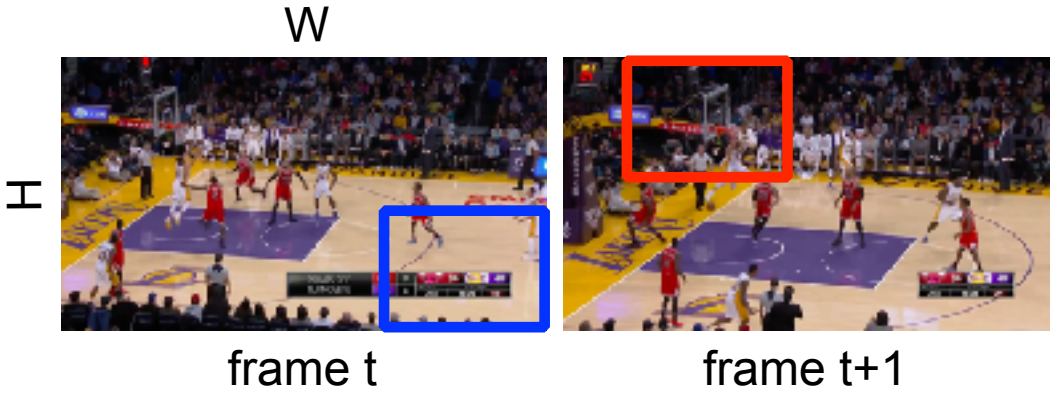
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

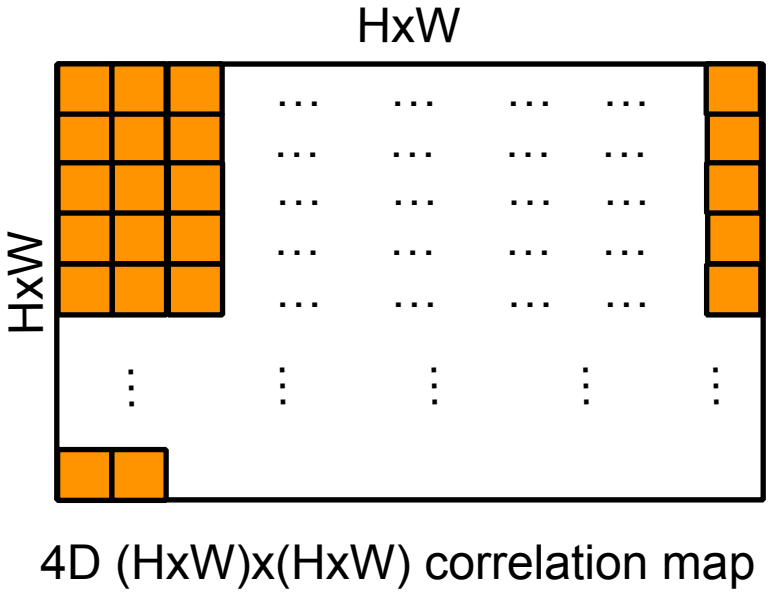




Correlation Layer

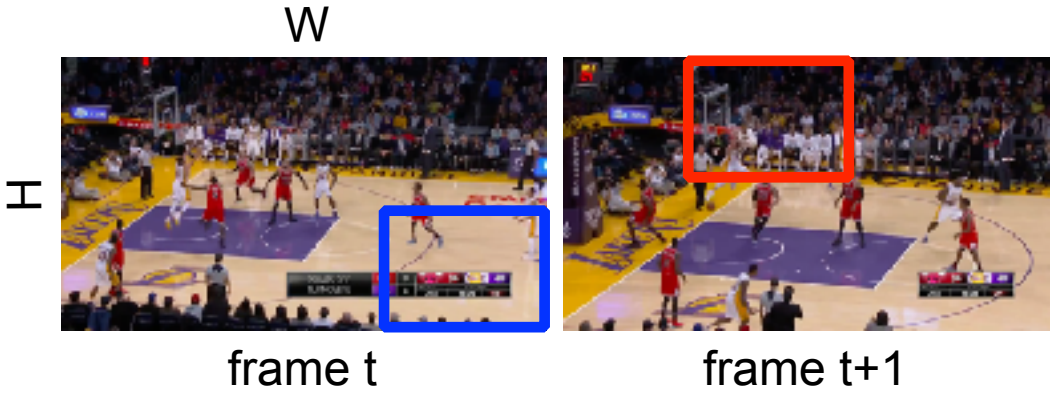
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

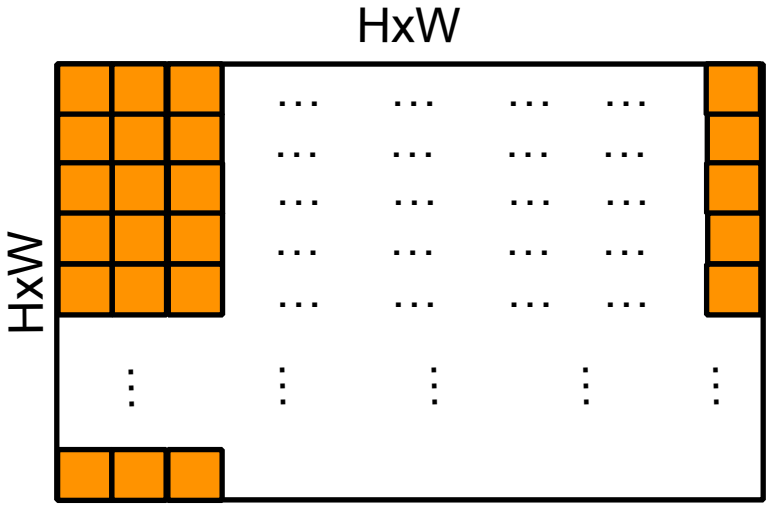




Correlation Layer

- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

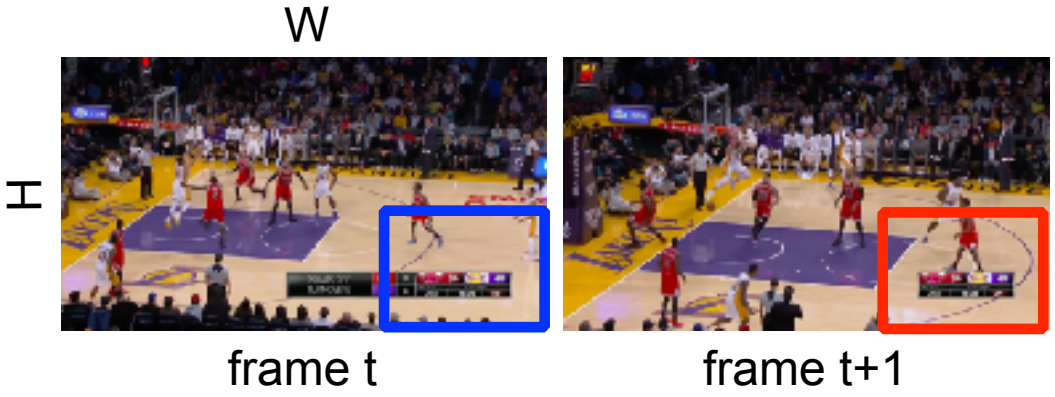


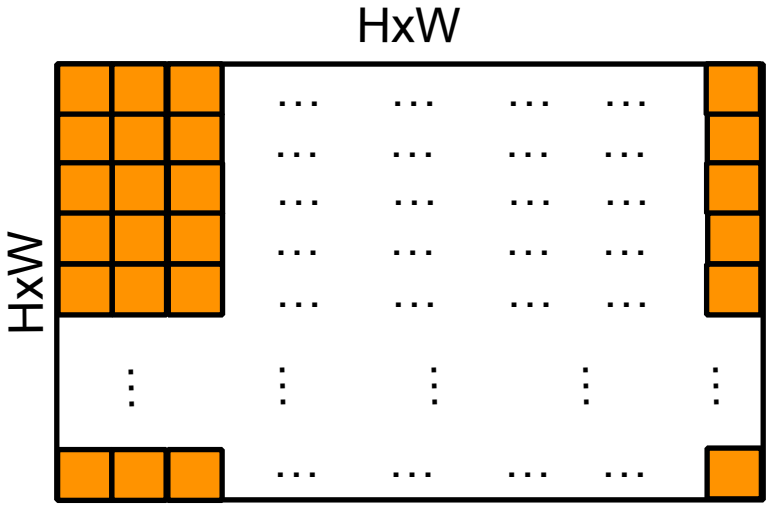


4D (HxW)x(HxW) correlation map

Correlation Layer

- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.





4D (HxW)x(HxW) correlation map

Correlation Layer

- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

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

- **This requires d · K^2 multiplications where d is the number of channels.**
- **Comparing all patch combinations involves W^2 · H^2 such computations.**

Correlation Layer

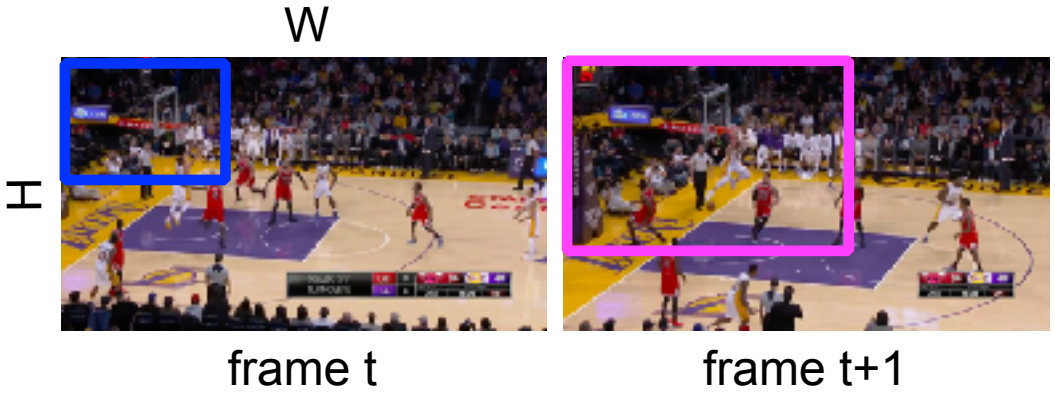
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

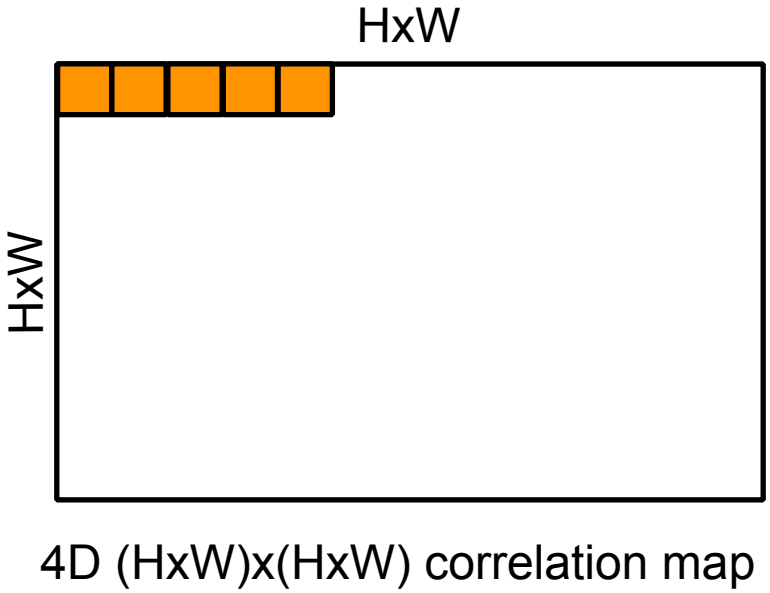
$$c(\mathbf{x}_1, \mathbf{x}_2) = \sum_{\mathbf{o} \in [-k,k] \times [-k,k]} \langle \mathbf{f}_1(\mathbf{x}_1 + \mathbf{o}), \mathbf{f}_2(\mathbf{x}_2 + \mathbf{o}) \rangle$$

Given a maximum displacement D, we compute correlations c(x1, x2) only in a neighborhood of size D = 2r + 1.

Correlation Layer

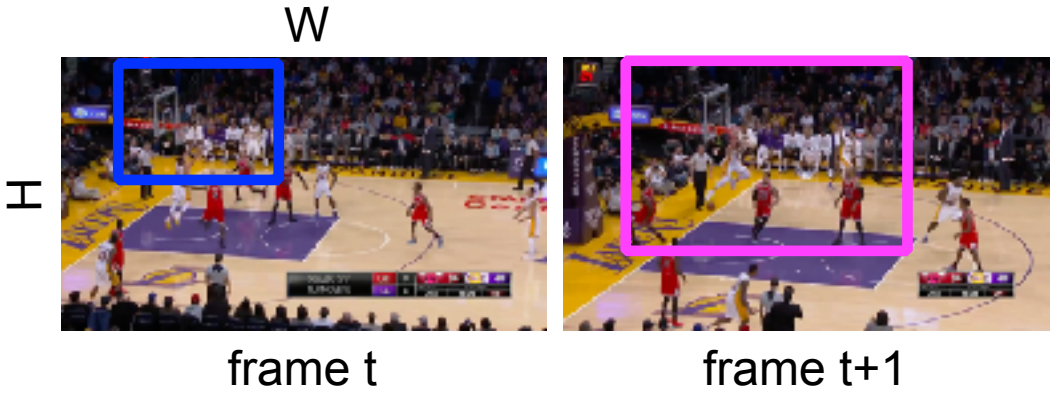
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

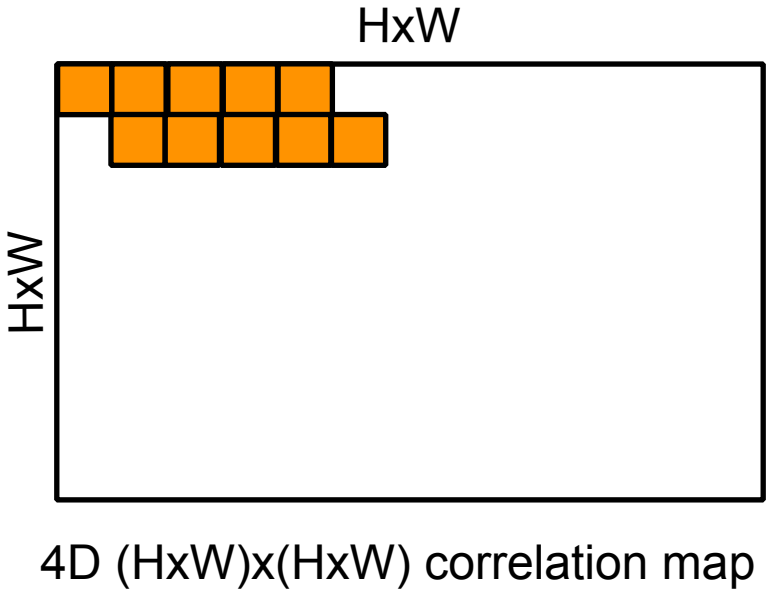




Correlation Layer

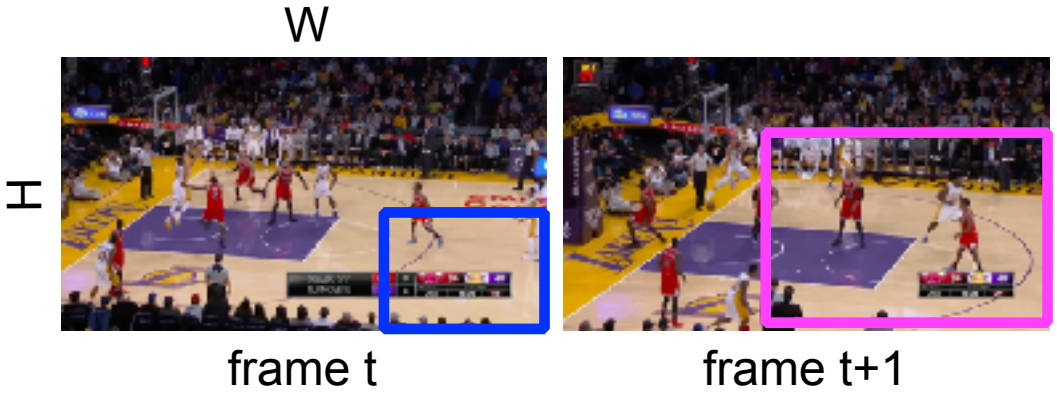
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

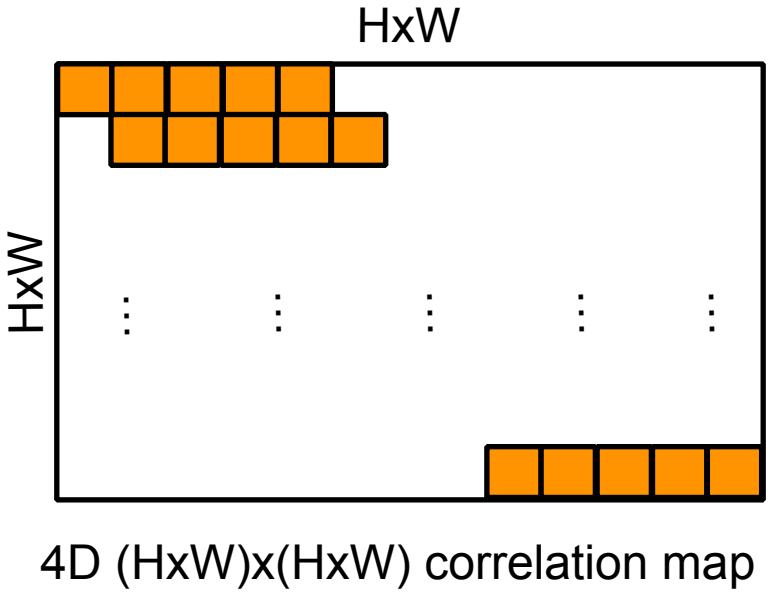




Correlation Layer

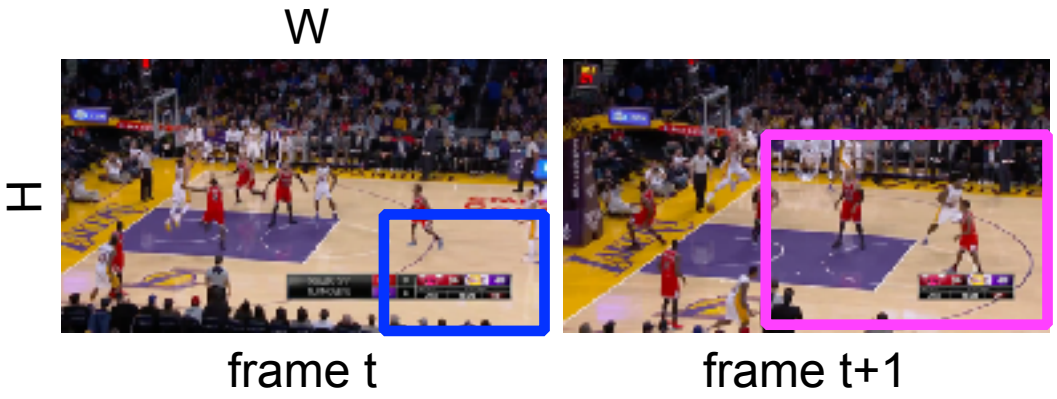
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.

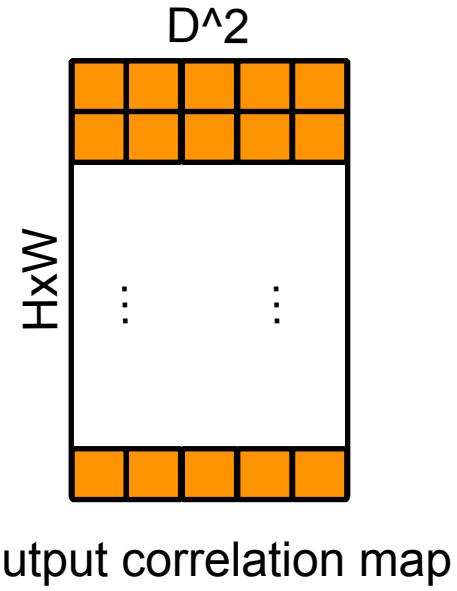




Correlation Layer

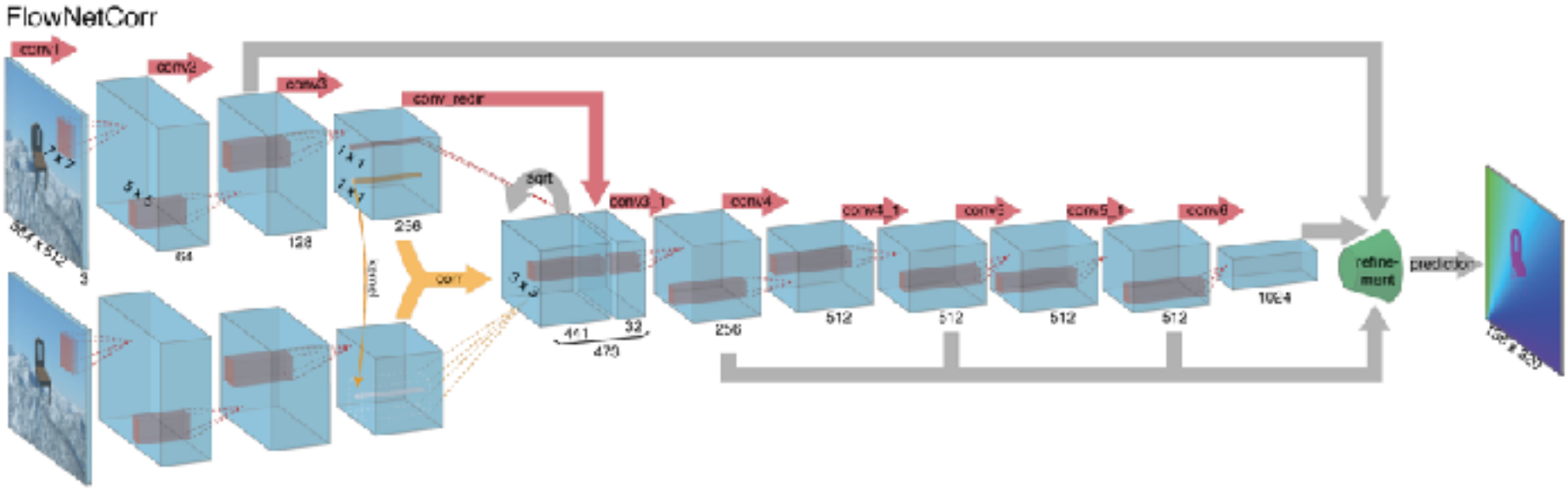
- Given two feature maps f1 and f2, the network compares each patch from f1 with each patch from f2 (using square patches of size K=2k+1)
- Correlation operator is identical to convolution, but instead of convolving data with a filter, it convolves data with other data.





FlowNetC (Correlation Network)

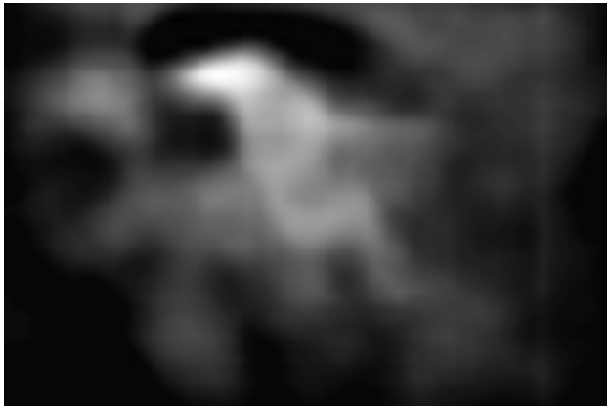
- Create two separate, yet identical processing streams for the two images.
- Combine them at a later stage via a correlation layer.



Feature Refinement

- 2D pooling reduces spatial resolution, which degrades performance for pixel-prediction tasks (e.g., optical flow).
- Deconvolutional and max unpooling layers are used to upscale feature maps to higher spatial resolution.

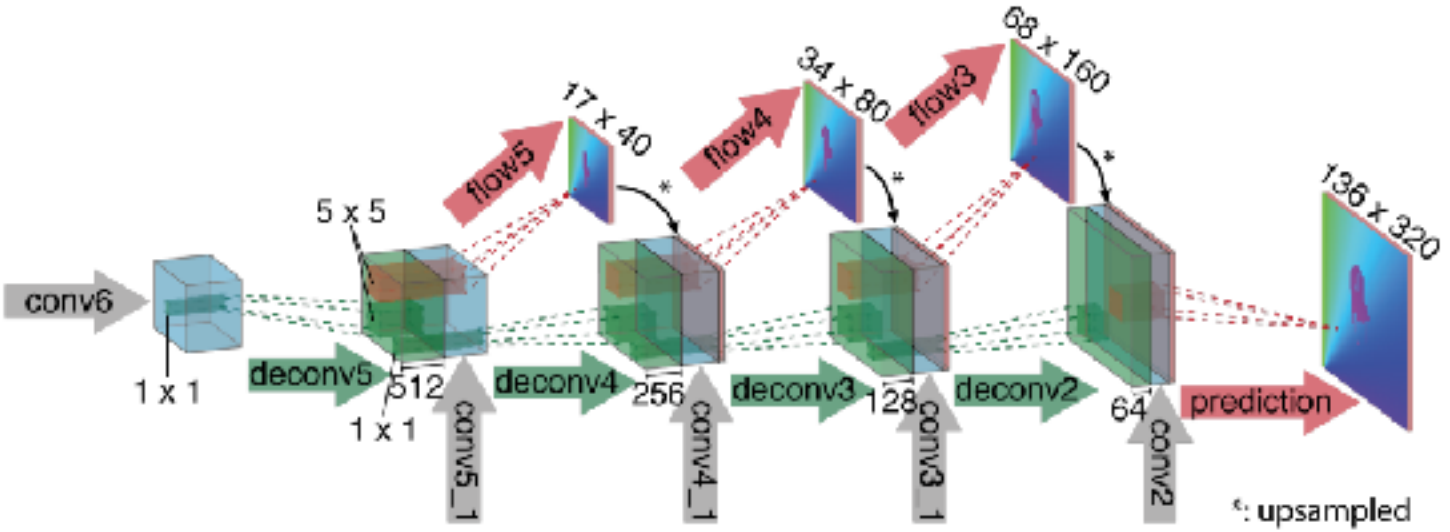
Input Image



Spatial Features in the Last Convolutional Layer

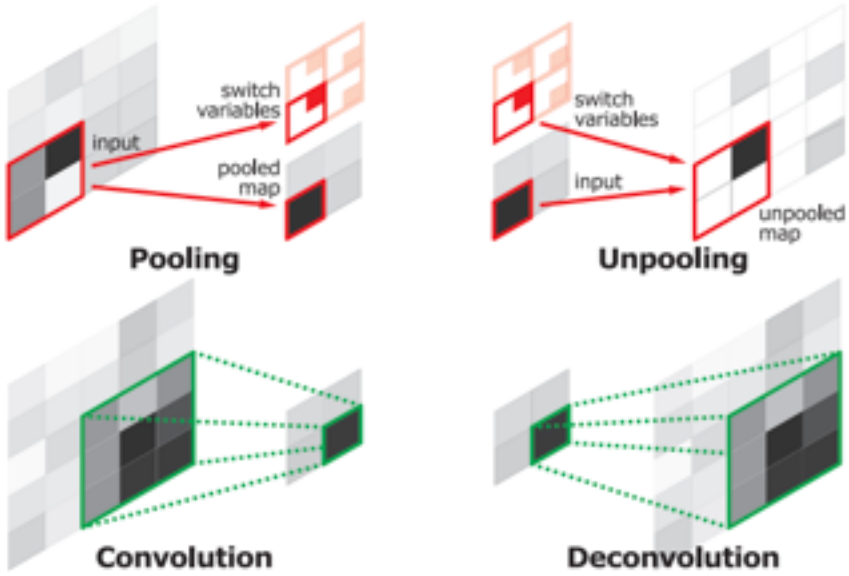
Feature Refinement

- 2D pooling reduces spatial resolution, which degrades performance for pixel-prediction tasks (e.g., optical flow).
- Deconvolutional and max unpooling layers are used to upscale feature maps to higher spatial resolution.



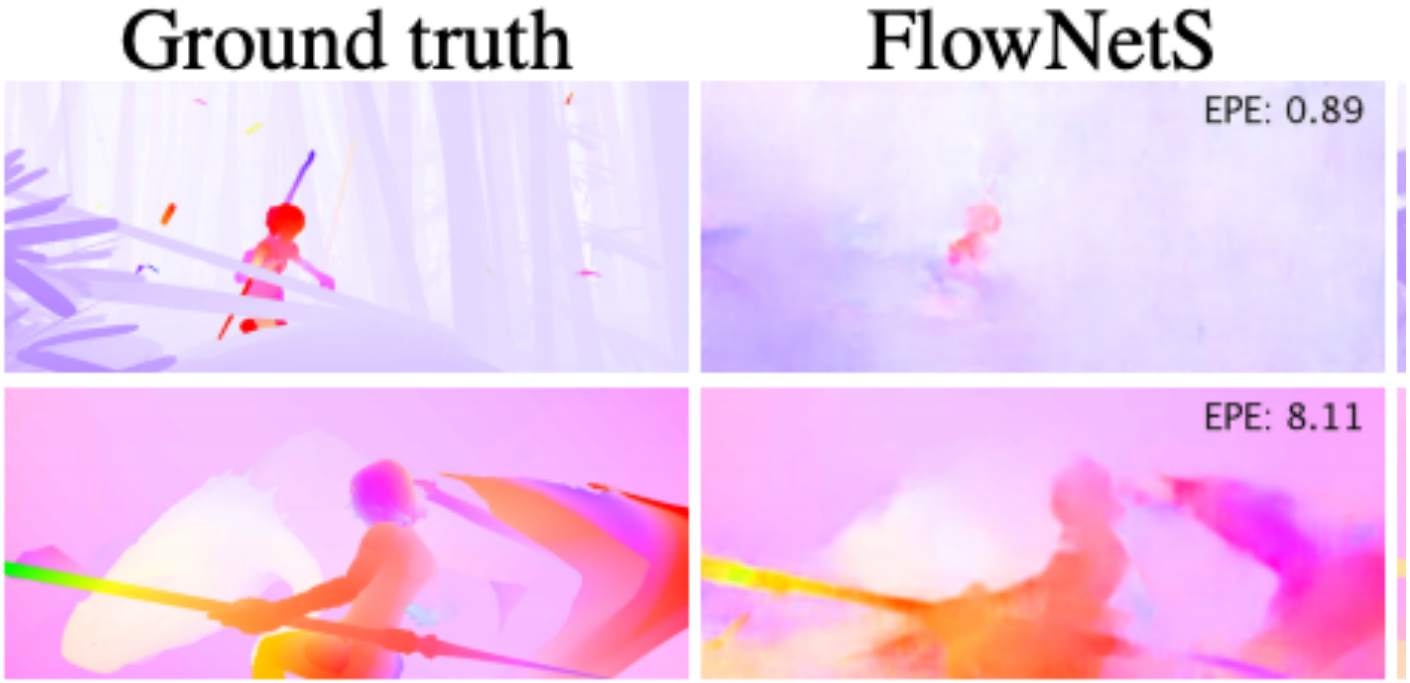
Feature Refinement

- The output of the deconvolutional and unpooling layers is an enlarged and dense activation map.



Loss Function

- The authors use Euclidean distance between the predicted flow vector and the ground truth, averaged over all pixels



Training Data

- The existing optical flow datasets are too small for effective CNN training.

	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%

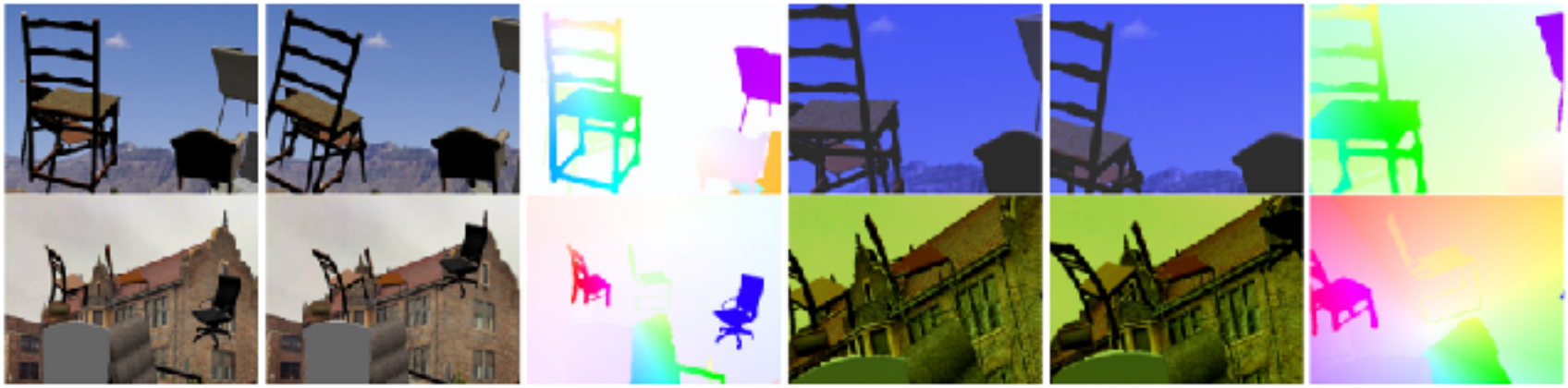
Training Data

- The existing optical flow datasets are too small for effective CNN training.

	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%
Flying Chairs	22,872	22,872	100%

Flying Chairs Dataset

- A synthetic dataset created by applying affine transformations to Flickr images and a rendered set of 3D chair models.
- 964 images from Flickr with a resolution of 1024 × 768 are used.
- 3D chairs are superimposed on the Flickr images.



Results

- The performance is evaluated using average endpoint errors (in pixels).

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]	2.40	4.12	3.70	6.29	3.47	3.8	0.31	3.24	0.39	3.55	2.94	16	-
DeepFlow [35]	3.31	5.38	4.56	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]	4.29	7.56	6.42	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

Results

- The performance is evaluated using average endpoint errors (in pixels).

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]	2.40	4.12	3.70	6.29	3.47	3.8	0.31	3.24	0.39	3.55	2.94	16	-
DeepFlow [35]	3.31	5.38	4.56	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]	4.29	7.56	6.42	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

The networks trained just on the non-realistic Flying Chairs perform very well on real optical flow datasets.

Results

- The performance is evaluated using average endpoint errors (in pixels).

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]	2.40	4.12	3.70	6.29	3.47	3.8	0.31	3.24	0.39	3.55	2.94	16	-
DeepFlow [35]	3.31	5.38	4.56	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]	4.29	7.56	6.42	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

FlowNetC performs very similarly to FlowNetS in most cases (on the realistic flow datasets).

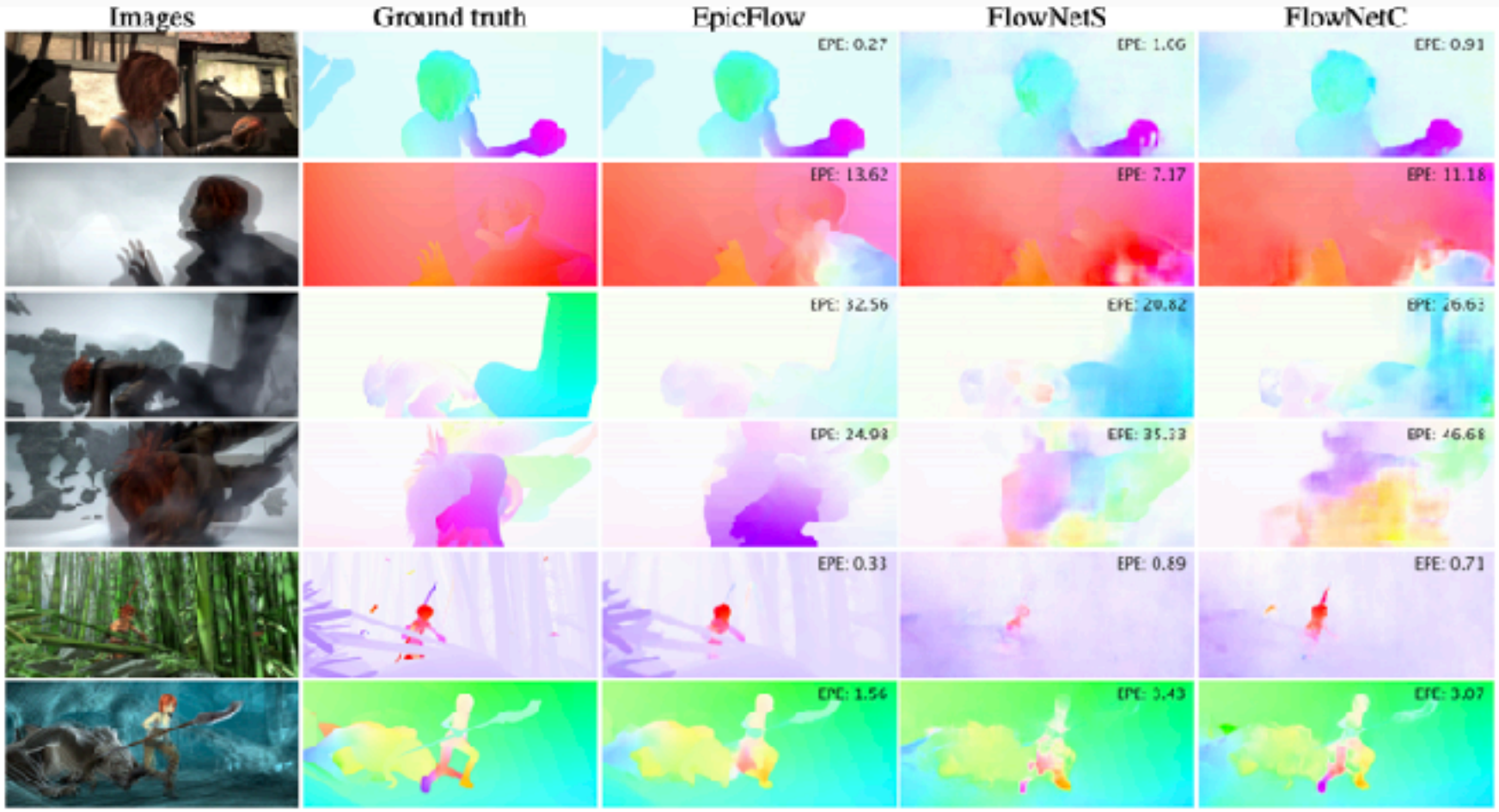
Results

- The performance is evaluated using average endpoint errors (in pixels).

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]	2.40	4.12	3.70	6.29	3.47	3.8	0.31	3.24	0.39	3.55	2.94	16	-
DeepFlow [35]	3.31	5.38	4.56	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]	4.29	7.56	6.42	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

Comparable results with state-of-the-art and lower computational cost.

Qualitative Results



Contributions

- The first attempt to train a CNN to directly predict optical flow from two input images.
- Simple, efficient, and effective approach.
- The introduction of Flying Chairs, the largest optical flow dataset at the time.
- Strong generalization from synthetic to real data.

Discussion Questions

- If we trained a video classification model on synthetic data and then tested it on real data it would perform poorly. Why does such a strategy work in this case?

Discussion Questions

- If we trained a video classification model on synthetic data and then tested it on real data it would perform poorly. Why does such a strategy work in this case?

- Why does the correlation network perform so similarly to the baseline network?

Discussion Questions

- If we trained a video classification model on synthetic data and then tested it on real data it would perform poorly. Why does such a strategy work in this case?
- Why does the correlation network perform so similarly to the baseline network?
- The FlowNets often produce visually appealing results, but their results are still worse in terms of endpoint error. Why?