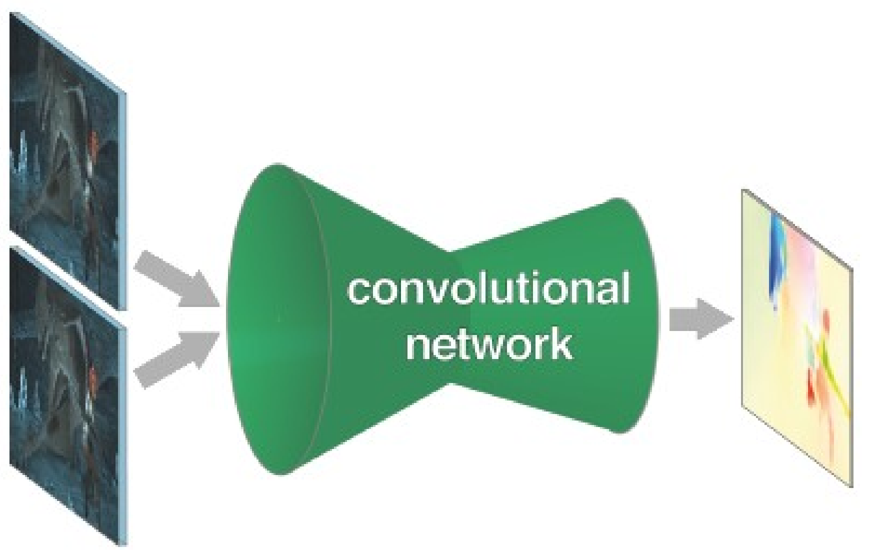
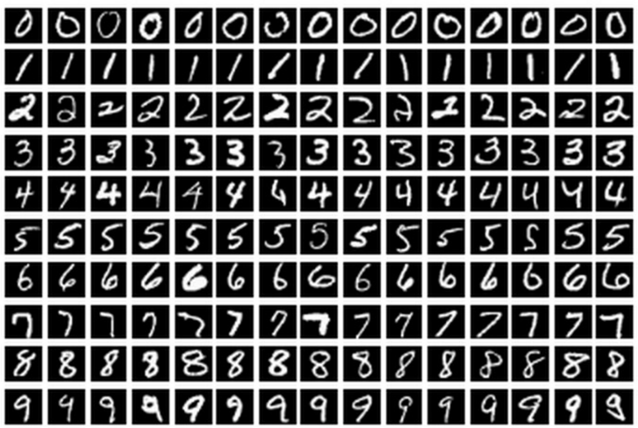
FlowNet: Learning Optical Flow with Convolutional Networks



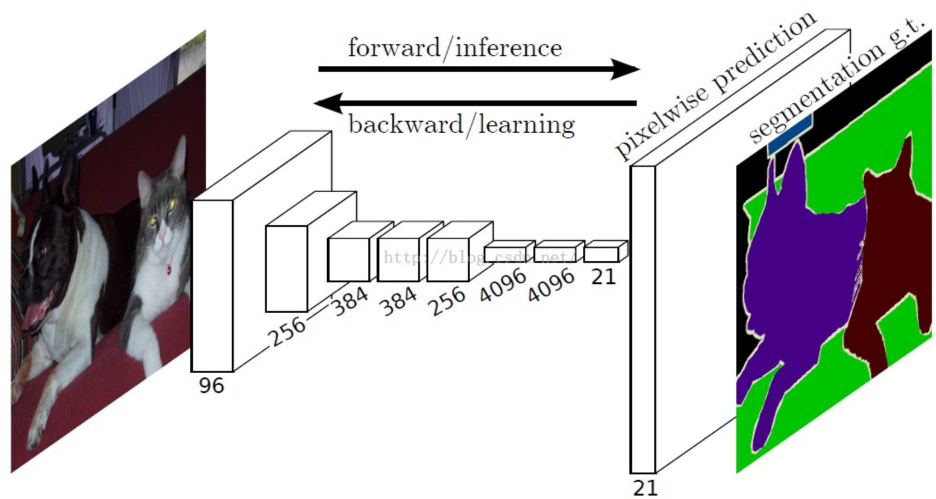
Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg , et al

Introduction

Classical application → Per-Pixel Prediction



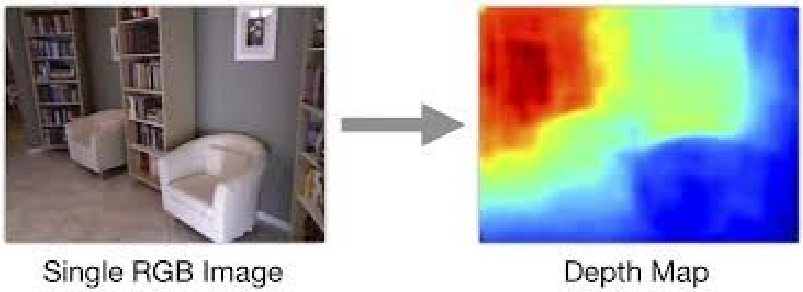
Handwritten Zip Code Recognition



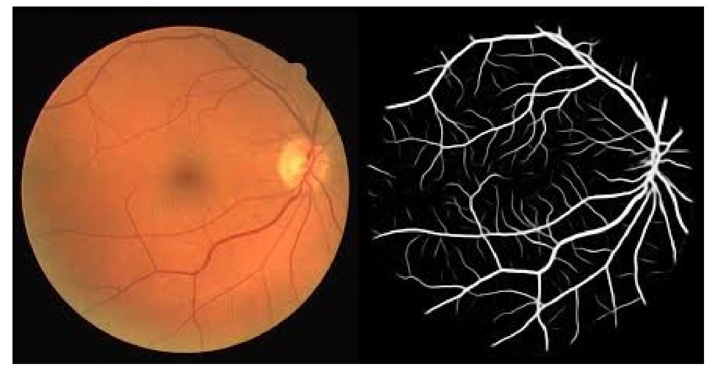
Fully convolutional networks for semantic segmentation

http://blog.csdn.net/tangwei2014/article/details/46882257
https://godelescherblog.wordpress.com/2015/09/20/numbers-exist-right/

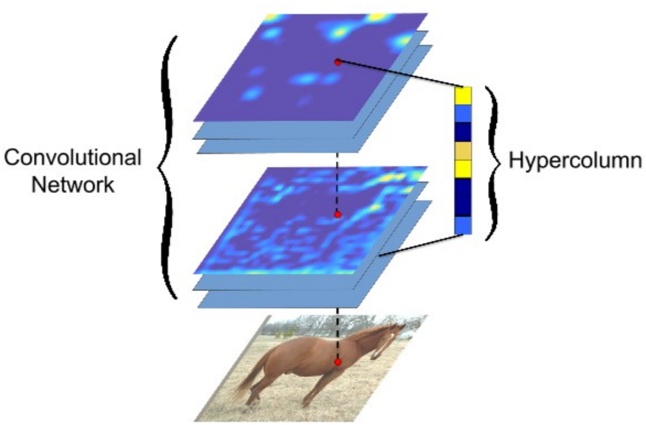
Other applications



Depth Estimation from Single Image



Edge detection



Key Point prediction

D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network.

Motivation

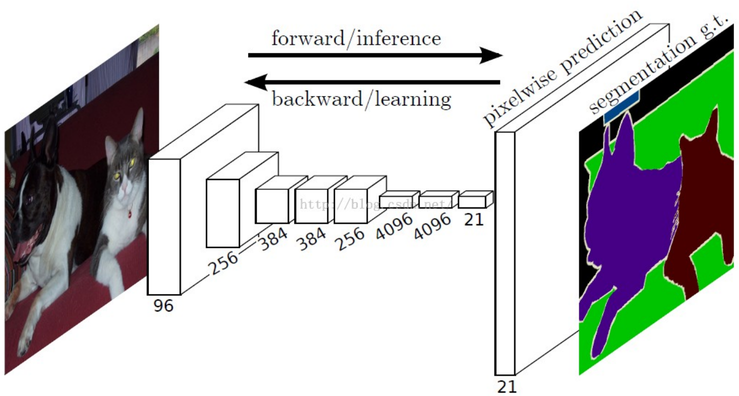
Two simple solutions

- Sliding Window
- Works well
- Drawbacks:
  - High Computational cost (re-use feature map)
  - Global Output Property(Sharp Edge)

- Upsample feature map to full resolution
- Stack all feature maps together
- Concatenated per-pixel feature vector to predict the value of interest

Motivation

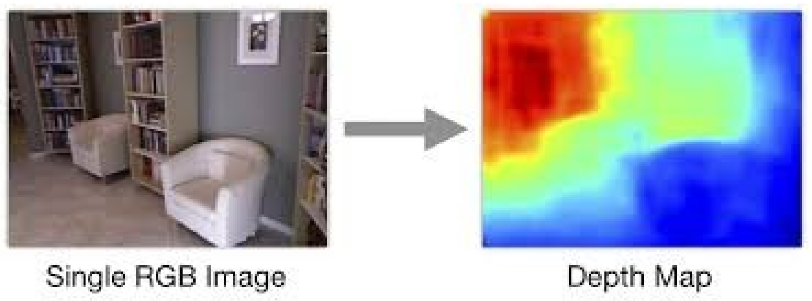
Two simple solutions



Fully Convolutional Networks for Semantic Segmentation

“Upconvolution”

Allowing refine feature maps to original size



Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network

“Concatenation”

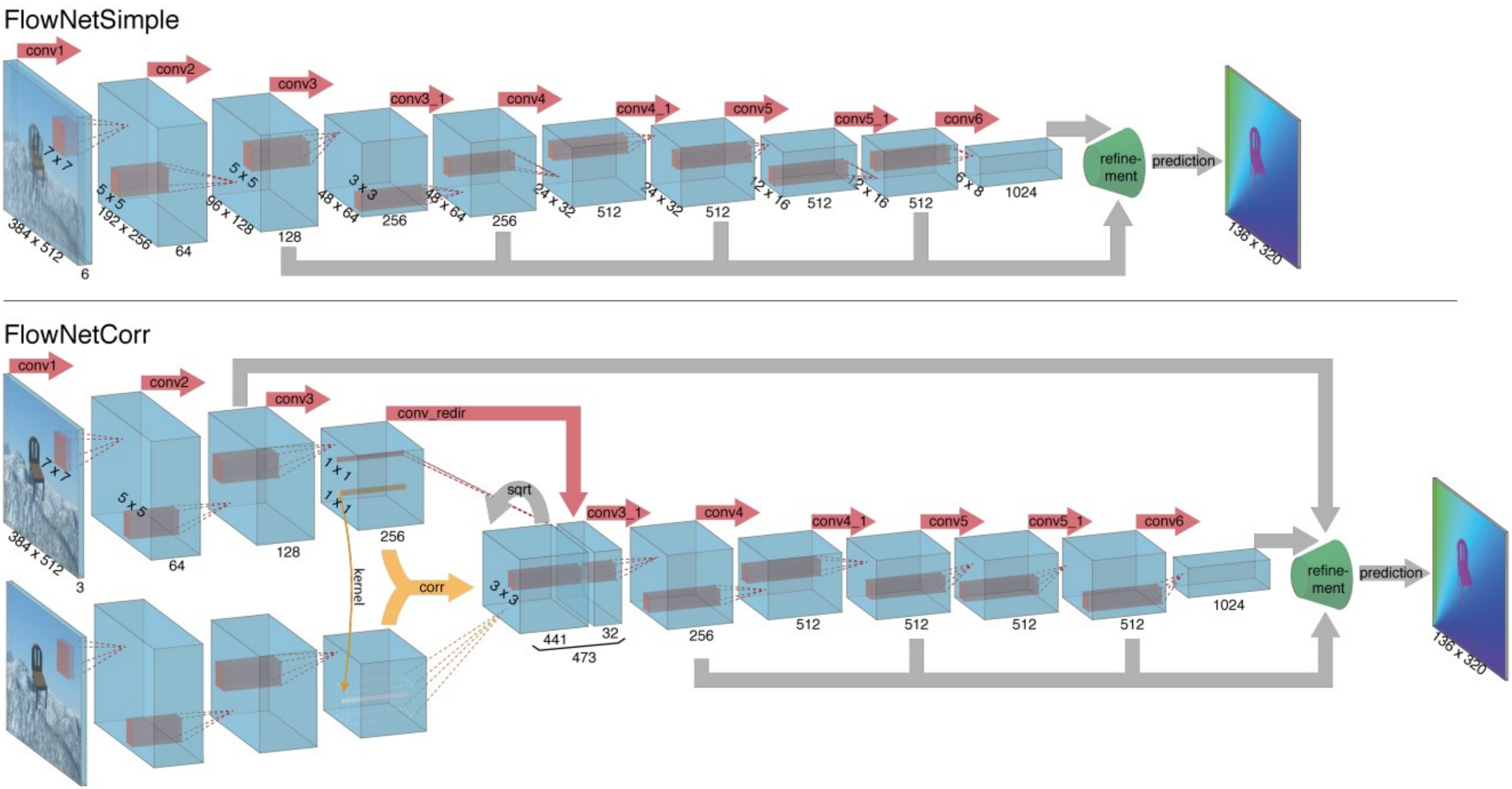
Concatenate coarse features with flow

Architecture

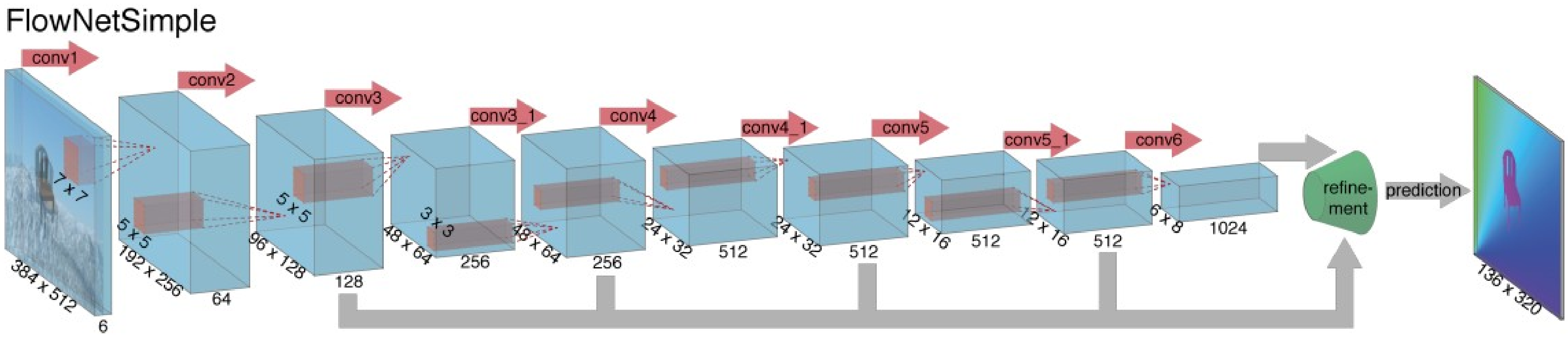
- FlowNetSimple
  - Generic network
  - Convolutinal layer only

  Drawbacks: Optimal point is not guaranteed
- FlowNetCorr
  - first produce meaningful representations of the two images separately
  - and then combine them on a higher level.

Architecture



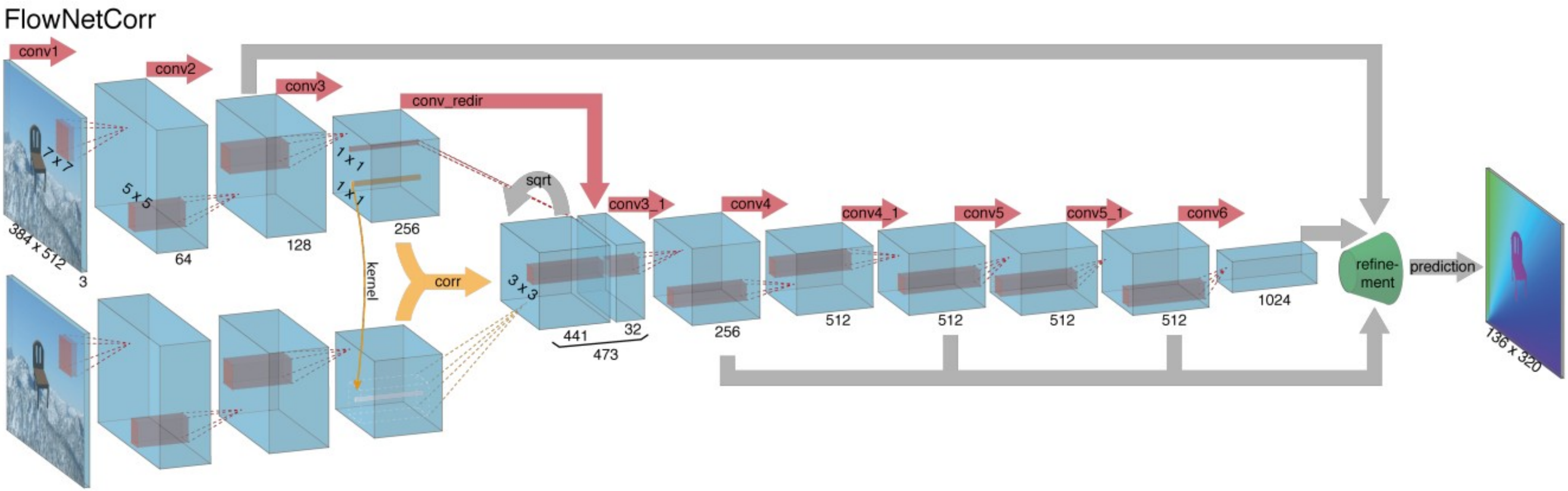
FlowNet Simple



Contractive Part:

- Overlay two input images
- 9 Convolutional layers (ReLu function each layer)
- Not fully connected network
- 6 layers have stride 2, double feature maps number
- 3 filters: 7*7, 5*5, 3*3

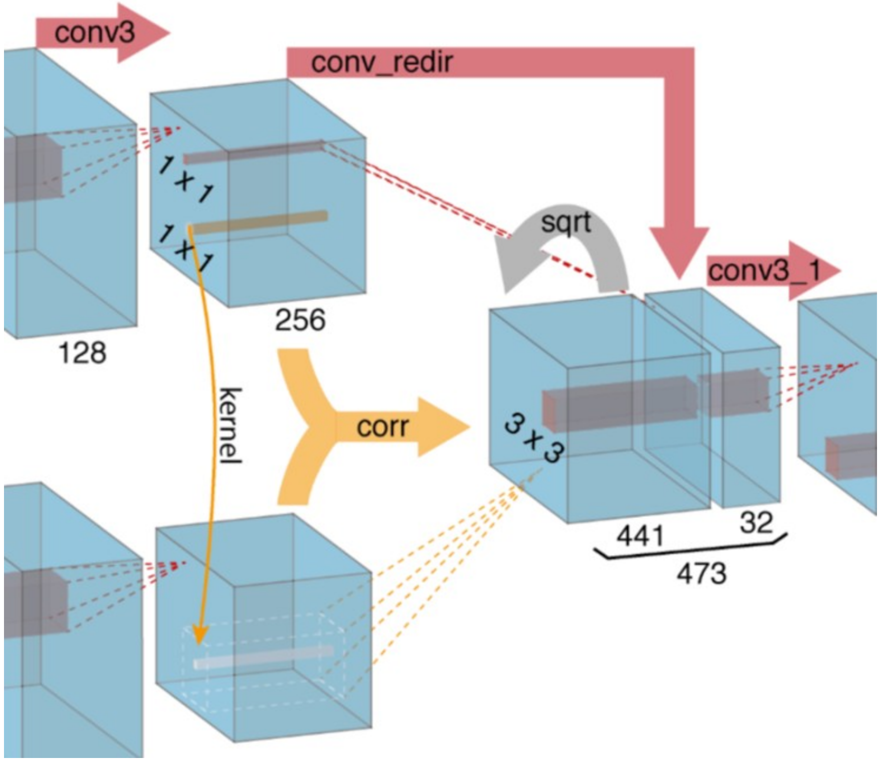
FlowNet Corr



Contractive Part:

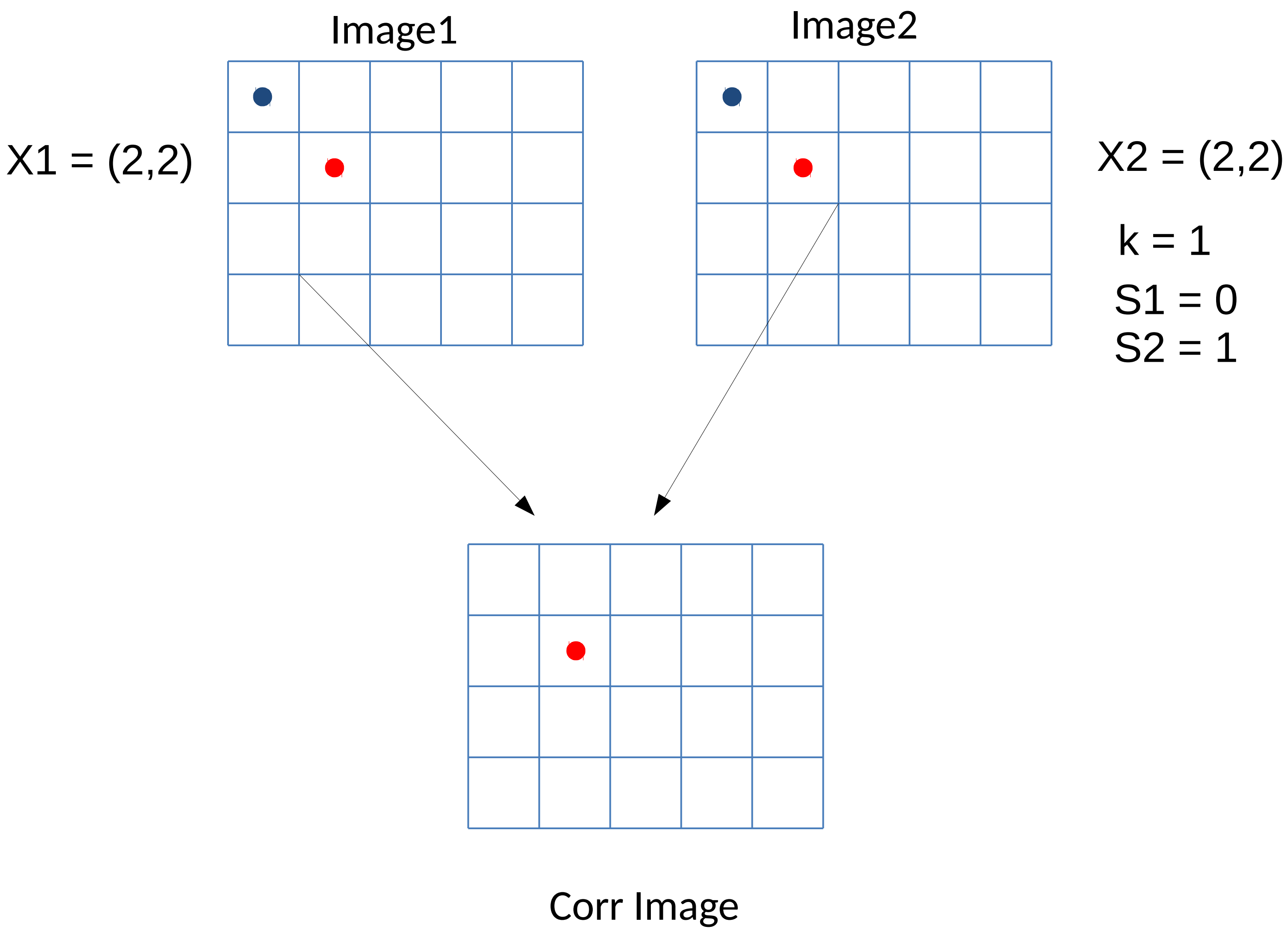
- Same architecture configuration as simple net
- First two layers are duplicated for each input
- Add in one more correlation layer at conv3

Correlation layer

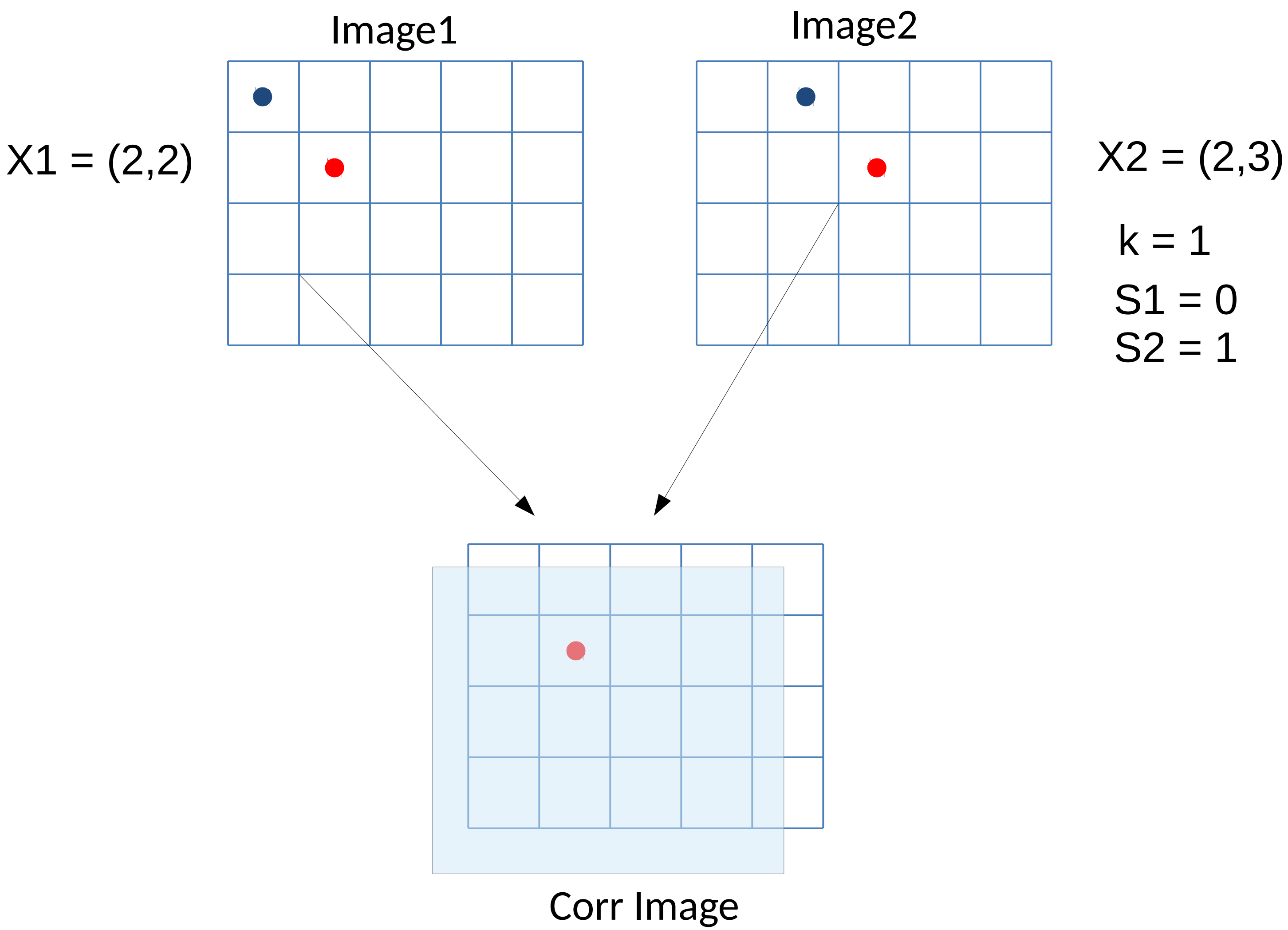


$$c(x_1, x_2) = \sum_{o \in [-k,k] \times [-k,k]} \langle f_1(x_1 + o), f_2(x_2 + o) \rangle$$

Correlation



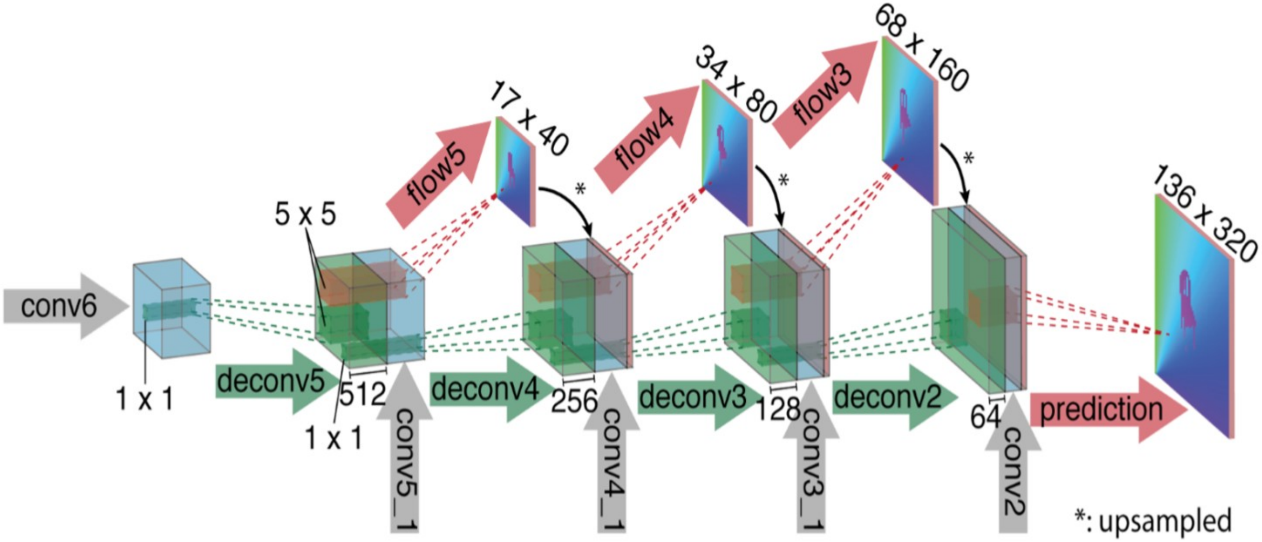
Correlation



Correlation layer

- Computing one patch $c(x_1, x_2)$ involves $c \times K^2$ where $K = 2k + 1$
- Computing all patches combination:
$$cw^2h^2K^2$$
- Limiting maximum displacement d at each x, therefore neighborhood window size:
$$D = 2d + 1$$
- Output size: $w \times h \times D^2$
- Computing expenses: $cw^2h^2K^2D^2$
- In practice, k=0, d = 20, s1= 1, s2 = 2

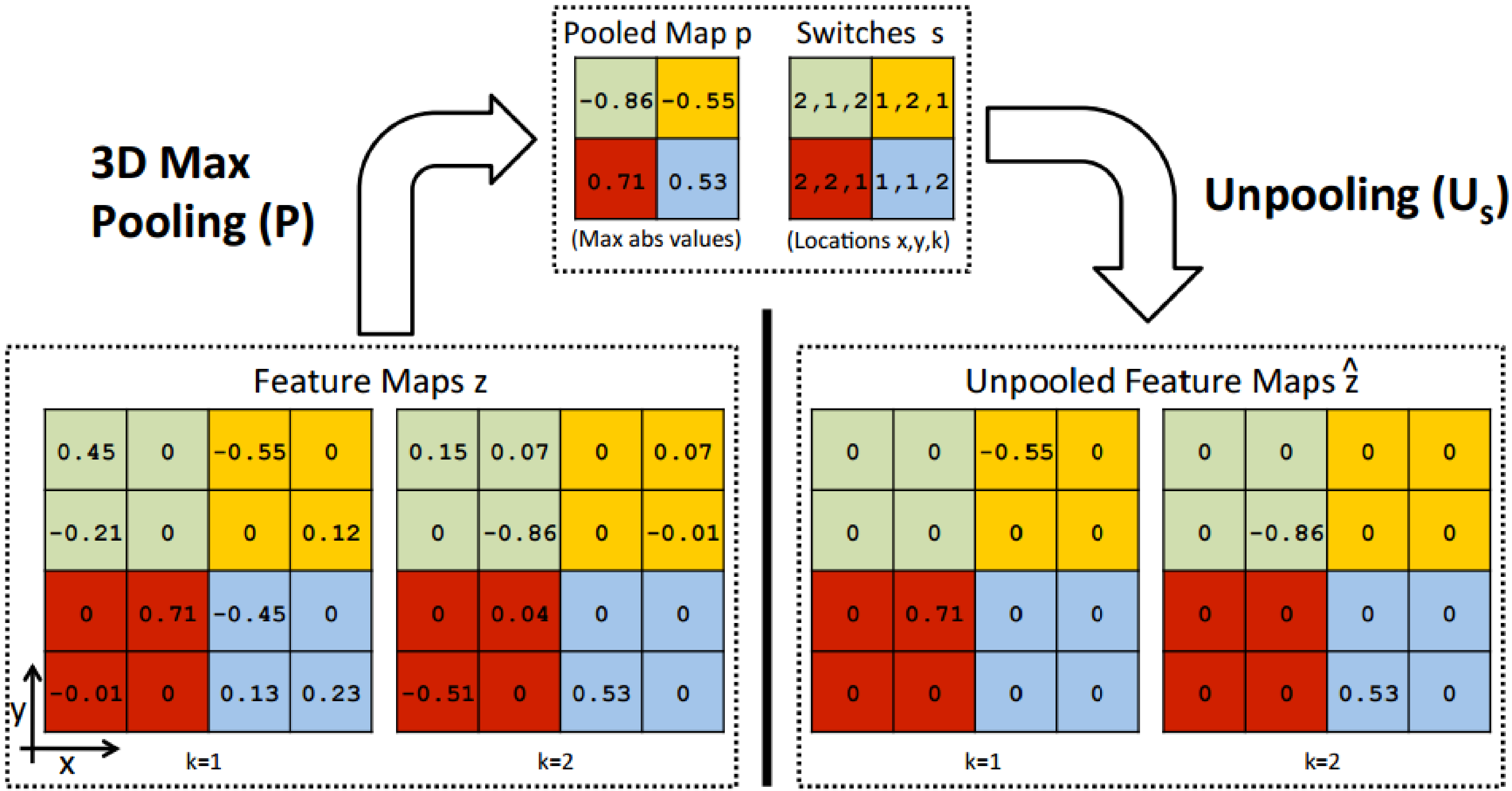
Refinement



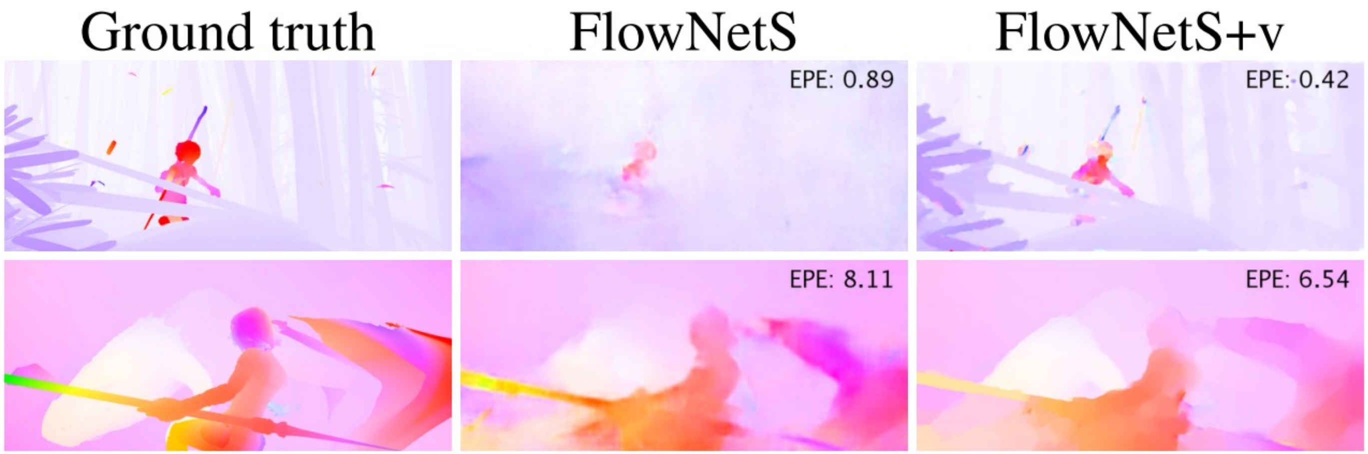
- Upconvolution = unpooling + convolution
- Cat(Upconvolution(feature map L) , feature map (L-1), upsampled flow map)
- Refinement rate: 5x, 4 hidden layers
- Last layer is bilinear upsampling without refinement

Unpooling

$$\hat{y}_1 = F_1 z_1 \qquad (3)$$



Alternative



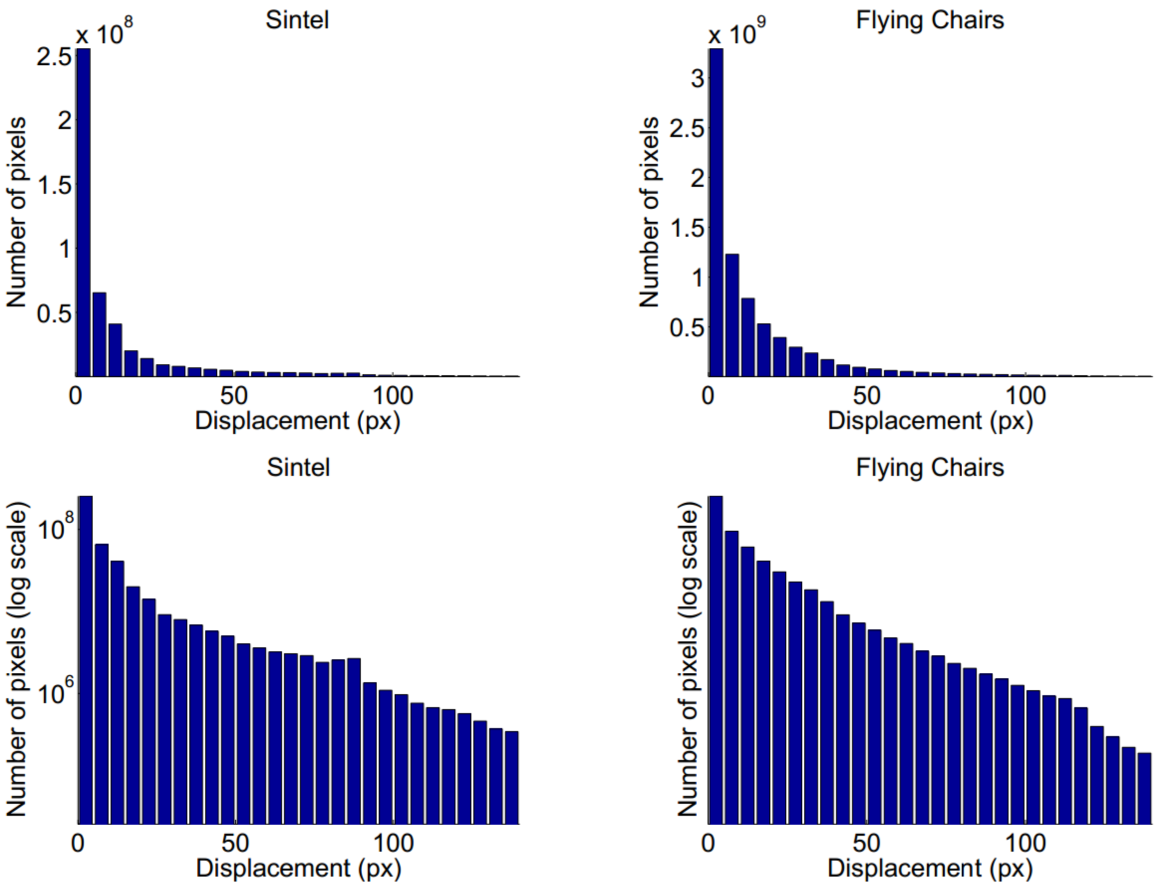
At last layer, use variational optical flow approach rather than bilinear upsampling
Steps:

1. At 4 times downsampled resolution feature map
2. Use coarse to fine scheme 20 iterations
3. Run 5 iterations at full resolution
4. Compute image boundary

This method computationally expensive but more accurate than bilinear upsample

Training Data Set

- **Middlebury**
  - 8 image pairs,
  - Displacement small, below 10 pixels
- **KITTI**
  - 194 training pairs,
  - only special motion types: moving observer,
  - lack ground truth of distant object
- **MPI Sintel**
  - 1041 training image pairs
  - Dense ground truth for small and large displacement
- **Flying Chair**
  - 22872 image pairs
  - With calculated flow fields



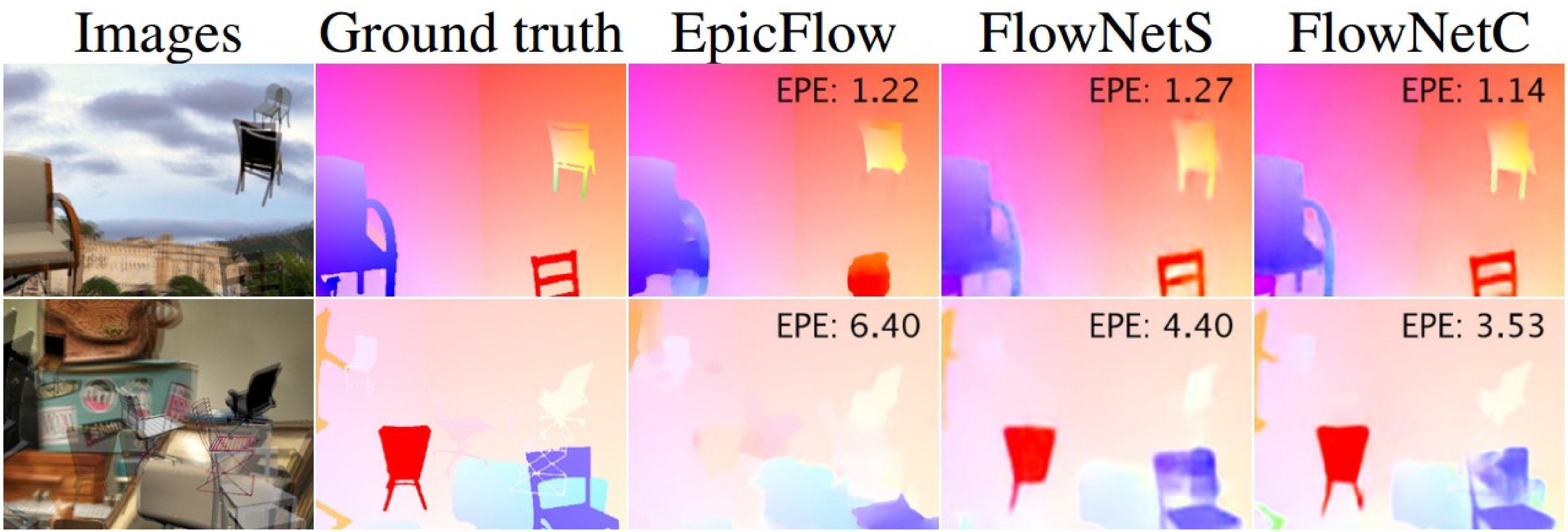
Data Augmentation
Affine Transformation: Translation, Rotation, Scaling,
Noise: Gaussian noise
Illumination: Brightness change, Contrast change, Gamma change, Color change

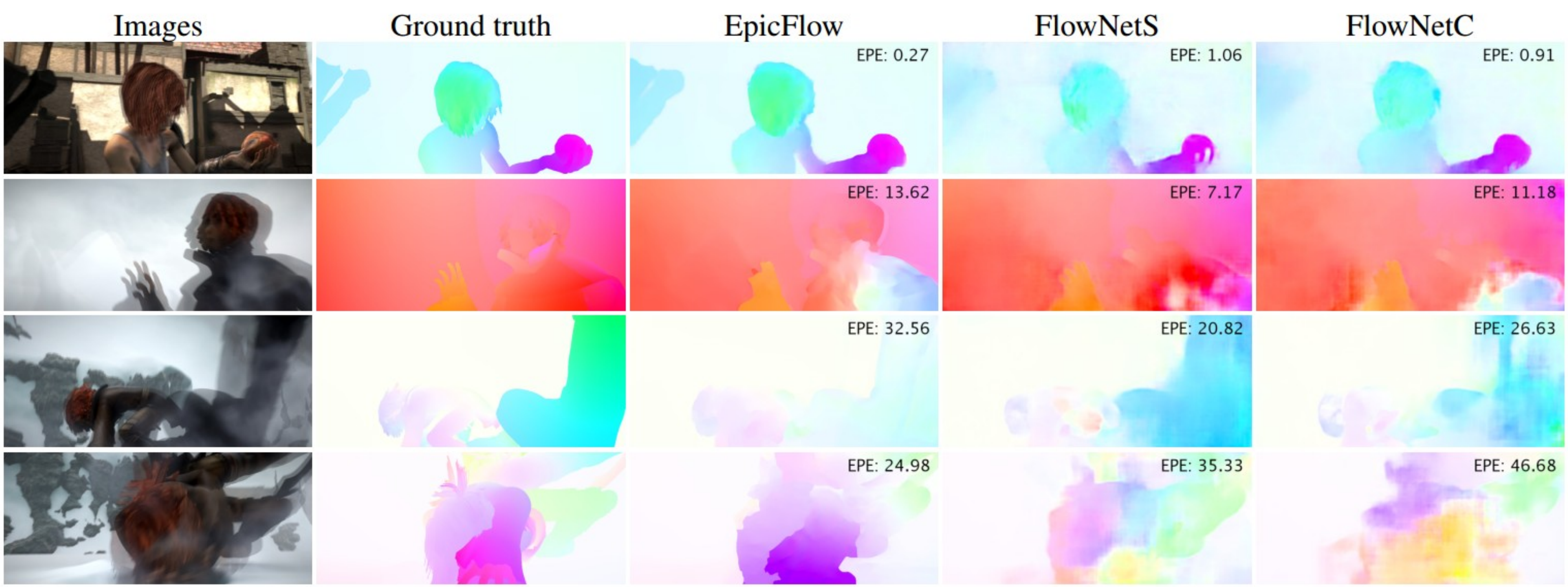
Results

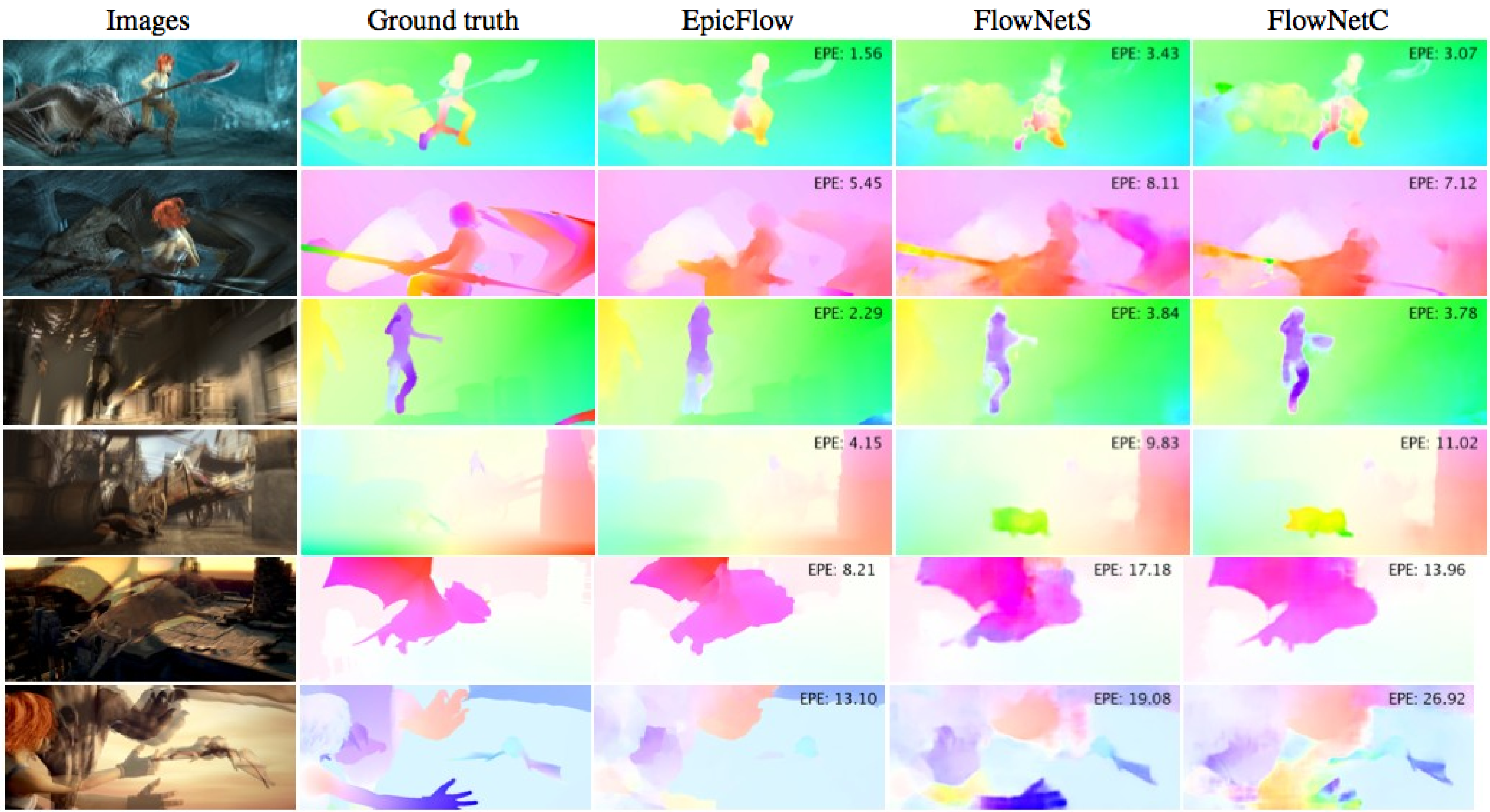
Method	Sintel Clean		Sintel Final		KITTI		Middlebury train		Middlebury test		Chairs	Time (sec)	
	train	test	train	test	train	test	AEE	AAE	AEE	AAE	test	CPU	GPU
EpicFlow [30]	2.40	4.12	3.70	6.29	3.47	3.8	0.31	3.24	0.39	3.55	2.94	16	-
DeepFlow [35]	3.31	5.38	4.56	7.21	4.58	5.8	0.21	3.04	0.42	4.22	3.53	17	-
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

Notes:

1. Before fine tuning, FlowNet outperforms LDOF, after fine tuning on Sintel, it outperforms EPPM
2. Generally performance is lower than EpicFlow or DeepFlow on existing data set, but for the chairs data set, it outperforms all the other methods


Images
Ground truth
EpicFlow
FlowNetS
FlowNetC
EPE: 1.22
EPE: 1.27
EPE: 1.14
EPE: 6.40
EPE: 4.40
EPE: 3.53


Images
Ground truth
EpicFlow
FlowNetS
FlowNetC
EPE: 0.27
EPE: 1.06
EPE: 0.91
EPE: 13.62
EPE: 7.17
EPE: 11.18
EPE: 32.56
EPE: 20.82
EPE: 26.63
EPE: 24.98
EPE: 35.33
EPE: 46.68


Images
Ground truth
EpicFlow
FlowNetS
FlowNetC
EPE: 1.56
EPE: 3.43
EPE: 3.07
EPE: 5.45
EPE: 8.11
EPE: 7.12
EPE: 2.29
EPE: 3.84
EPE: 3.78
EPE: 4.15
EPE: 9.83
EPE: 11.02
EPE: 8.21
EPE: 17.18
EPE: 13.96
EPE: 13.10
EPE: 19.08
EPE: 26.92

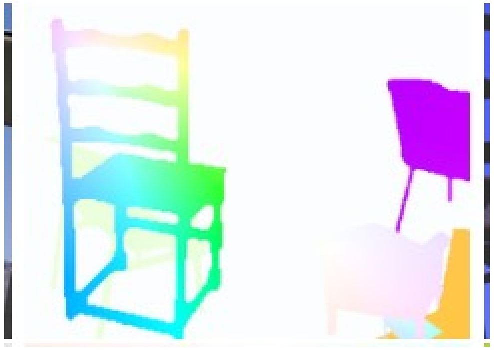
Analysis

Problems

- FlowNetS is better at generalization compared to FlowNetC. FlowNetC slightly more over-fits to training data
- The network can not understand the training samples, it somehow adapts more to the training data.
- FlowNetC is more sensitive to large displacement

Potential Solutions

- Data Augmentation is necessary! It helps to reduce end point error. Better training data will make FlowNet an advantage
- Maximum displacement of the correlation does not allow very large motion predictions. Increase maximum displacement at cost of computational efficiency



Q&A

Video

https://www.youtube.com/watch?v=k_wkDLJ8lJE