시각 Tech. 심화과정 (3)

연세대학교 컴퓨터과학과

김 선 주

SK New ICT Academy
- AI Course -

AGENDA

- Video Processing
- Geometry (Optical Flow, Stereo)

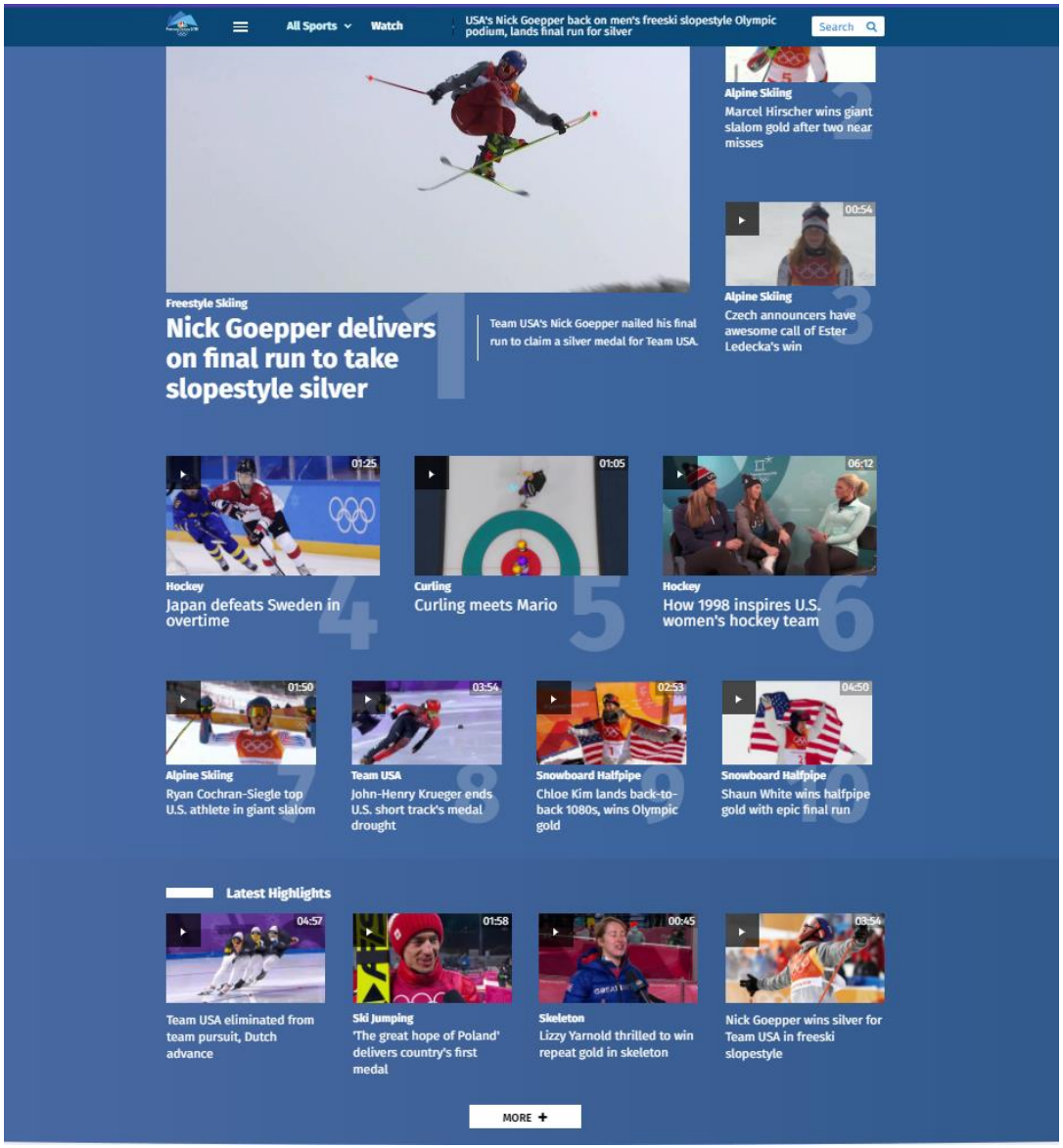
VIDEO PROCESSING

Semi-automatic Data Collection for Understanding Video with Fine-grained Motion (Baseball Video Dataset)

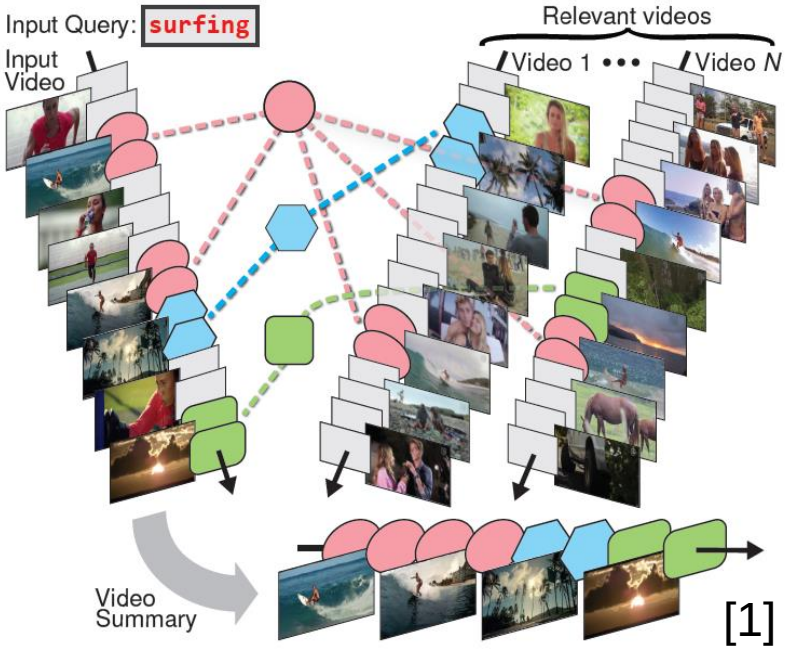
Yonsei University

Seon Joo Kim

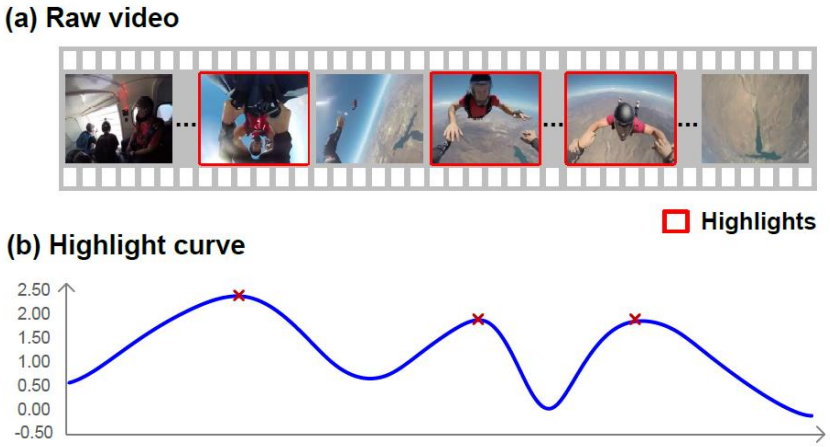
Sports Highlights



Video Summarization / Highlight Detection



[1]



[2]

[1] W. Chu et. al., "Video Co-summarization: Video Summarization by Visual Co-occurrence", CVPR 2015
[2] T. Yao et. al., "Highlight Detection with Pairwise Deep Ranking for First-Person Video Summarization", CVPR 2016

Query-based Highlights for Baseball Games

Input Video (sports broadcast)

Question

① Show me the scoring footages of team A from the game last night

② Show me all the homeruns from all pro baseball games from yesterday

user

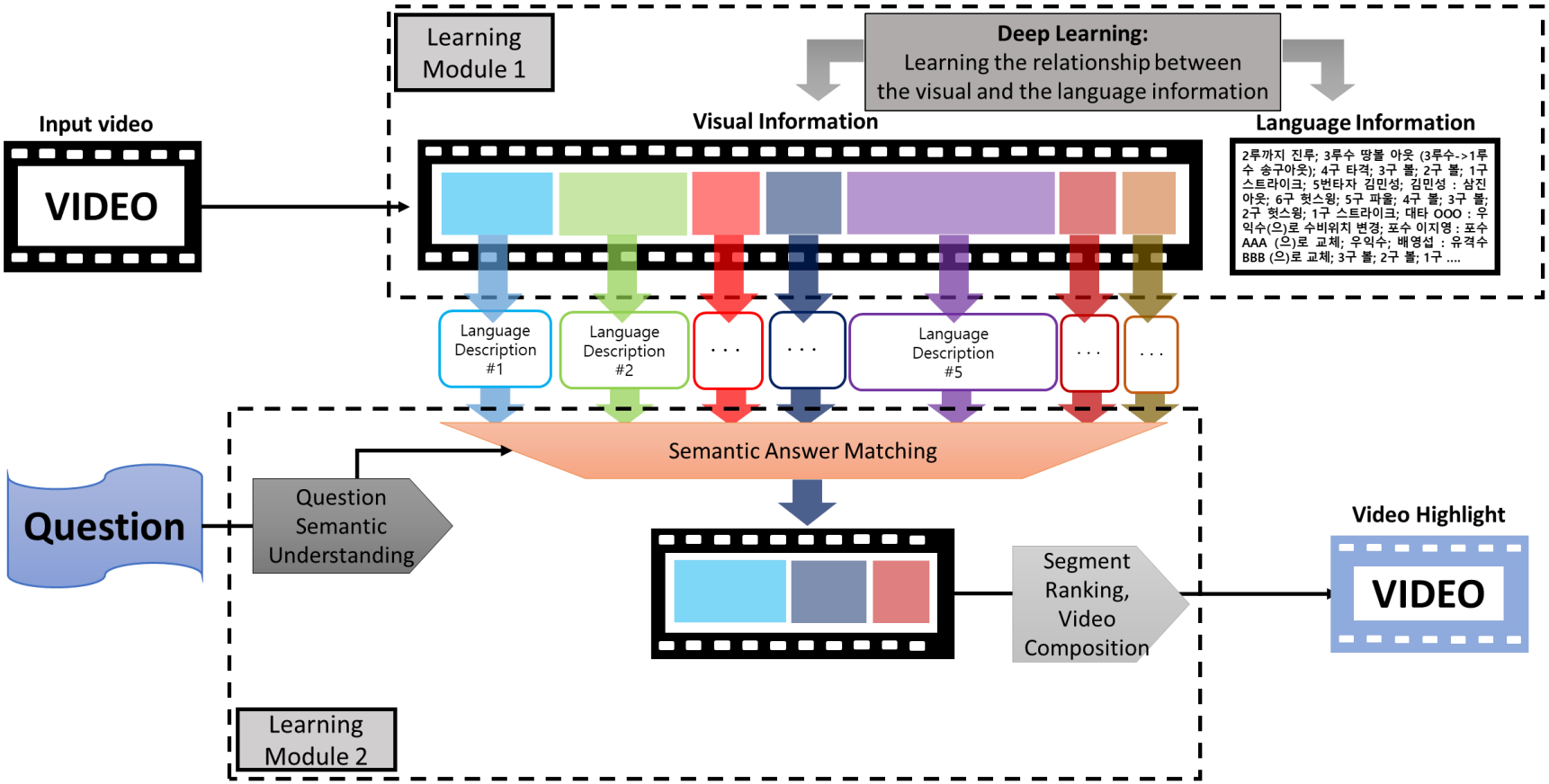
Generate Question-based Highlights

① Scoring highlights of team A

② Homerun collection highlights

Video Recognition, Video Highlight/Summarization, VQA, NLP

Query-based Highlights for Baseball Games



Understanding Baseball Games: Automatic Play-by-Play Generation

All Plays | Scoring Plays

Dodgers - Top 1st

SIMS PITCHING FOR ATL

Taylor flied out to center.

PITCH		TYPE	MPH
1	Strike Looking	Fastball	91
2	Strike Swinging	Fastball	92
3	Ball	Curve	73
4	Foul Ball	Slider	86
5	Ball	Fastball	92
6	Fly Out	Slider	88

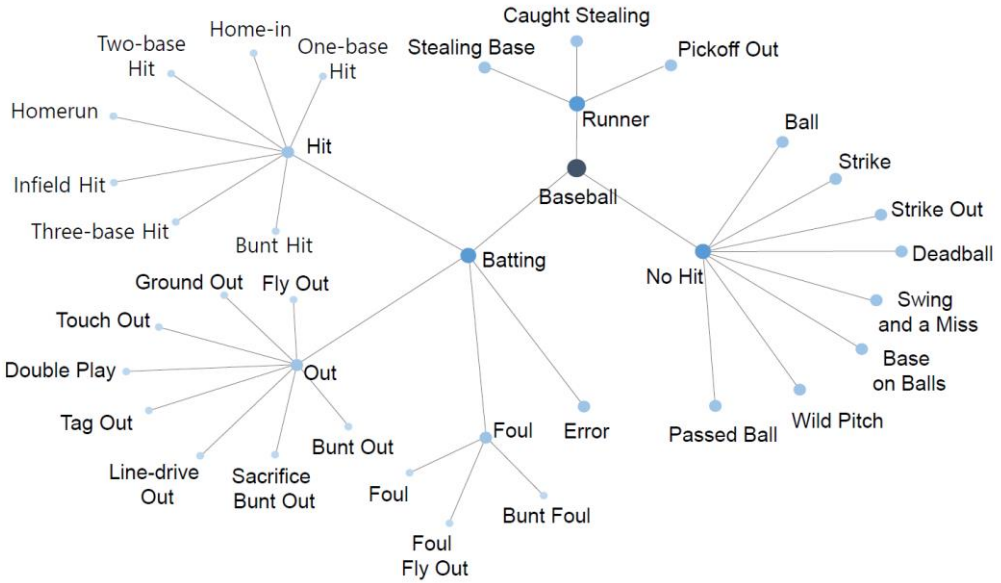
Seager popped out to shortstop.

Turner singled to right.

Bellinger lined out to center.

Dataset???

Baseball Video Database (BBDB)



1,172 games

4,254 hours of baseball

500 million frames

30 activity labels

Per game, we also have:

- Play-by-play text
- Highlight

Baseball Video Database (BBDB)

But how do label all those videos???

Semi-Automatic Labelling

vs. Other Video Dataset

Dataset	#instances†/#videos	Avg. Duration	Untrimmed	Detection	Sequential‡
UCF101 [32]	/13k	7s	-	-	-
HMDB51 [21]	/7k	3s	-	-	-
Kinetics [4]	/306k	10s	-	-	-
Sports1M [17]	/1.1m	300s	✓	-	-
Youtube8M [1]	/8.3m	230s	✓	-	-
THUMOS15 [13]	23.1k/21k	4s	✓	✓	△
MultiTHUMOS [38]	39k/400	270s	✓	✓	△
ActivityNet [3]	28k/20k	180s	✓	✓	△
Hollywood2 [23]	-/3.7k	20s	✓	✓	△
Cooking [27]	5.6k/44	600s	✓	✓	✓
Breakfast [20]	11k/2k	140s	✓	✓	✓
BBDB (ours)	**405k**/1k	**13,000s**	✓	✓	✓

IoU threshold	0.3	0.4	0.5	0.6	0.7
BBDB	98.2	95.5	89.0	68.7	49.5

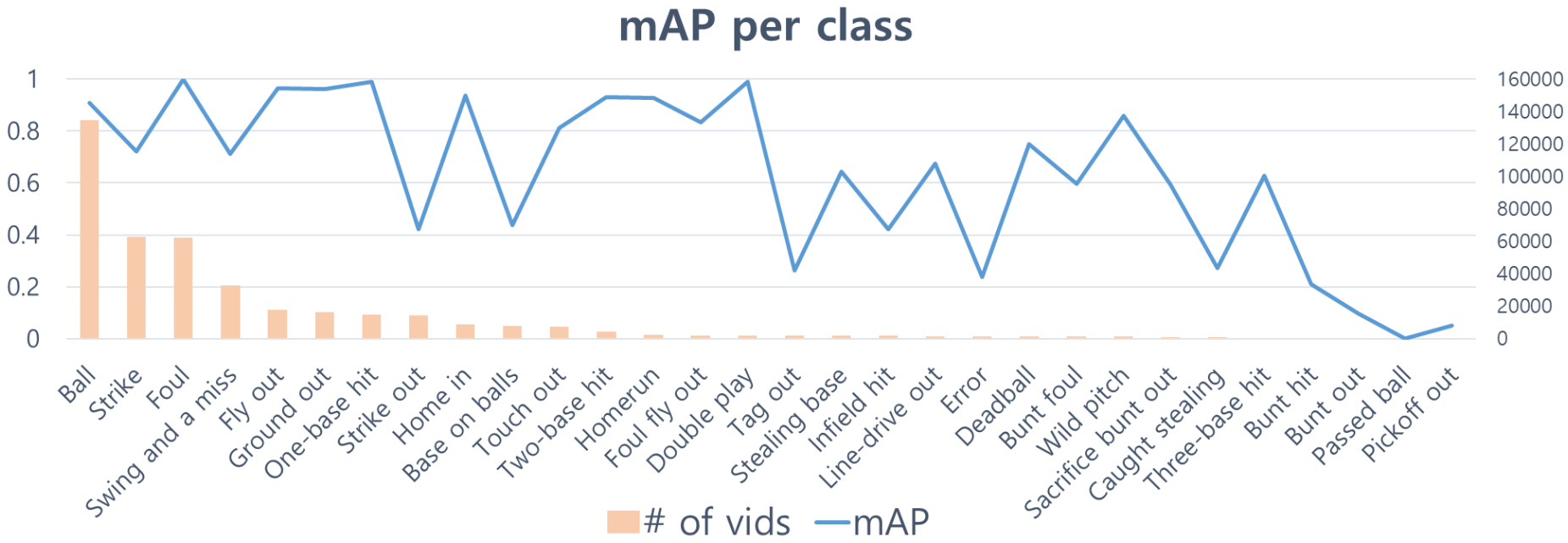
Table 1. Precision of semi-automatic labeling against human labeling on the BBDB. IoU threshold ranges from 0.3 to 0.7.

Class	count	mean	stdev
Foul fly out	8	9.16	0.96
Sacrifice bunt out	2	8.86	1.00
Double play	13	8.36	1.34
Deadball	23	6.61	1.36
Strike	333	5.45	1.44
Swing and a miss	141	5.45	1.44
⋮			
Wild pitch	6	7.50	4.12
Error	11	12.90	4.87
Home in	76	14.93	6.18
Homerun	16	17.38	8.97

Table 2. Clip length statistics on 6 manually annotated games. Count is the number of clips, and the unit of mean is second.

Algorithm Evaluation

Method	mAP	Cost
Action Recognition		
IDT + FV [36]	23.6	12 days
Single frame [31]	35.0	2 days
Optical flow stacking [30]	34.1	1 day
Two-stream (Avg) [30]	36.9	3 days
Two-stream (SVM) [30]	41.7	3 days
C3D [35]	40.2	18 hrs
CNN+GRU [8] (Oversampling)	**62.8**	2.5 days



Ball (True Positive)

Ground Truth : Ball
Predicted action (probability) : Ball (0.998)

Ground Truth : Ball
Predicted action (probability) : Ball (0.999)

Some Results

Ball (False Positive)

Ground Truth : Strike
Predicted action (probability) : Ball (0.999)

Ground Truth : Strike
Predicted action (probability) : Ball (0.696)

Hit by Pitch (True Positive)

Ground Truth : Hit by Pitch
Predicted action (probability) : Hit by Pitch (1.0)

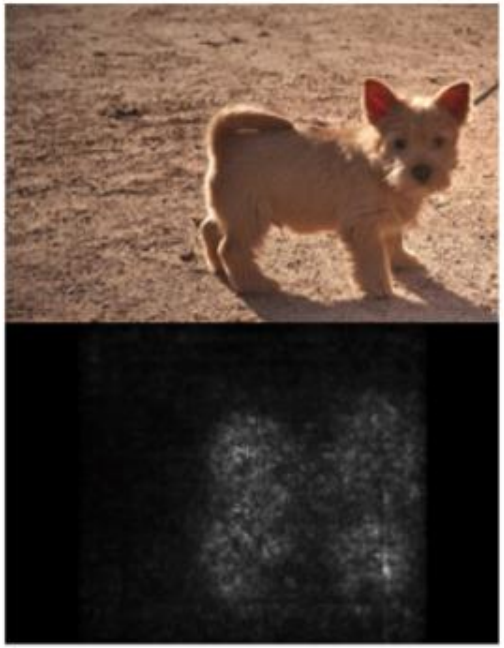
Ground Truth : Hit by Pitch
Predicted action (probability) : Hit by Pitch (1.0)

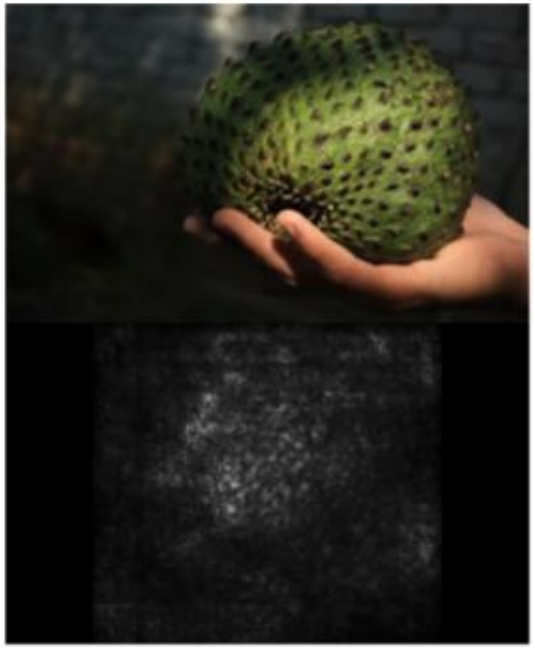
Hit by Pitch (False Positive)

Ground Truth : Foul
Predicted action (probability) : Hit by Pitch (0.999)

Ground Truth : Foul
Predicted action (probability) : Hit by Pitch (0.885)

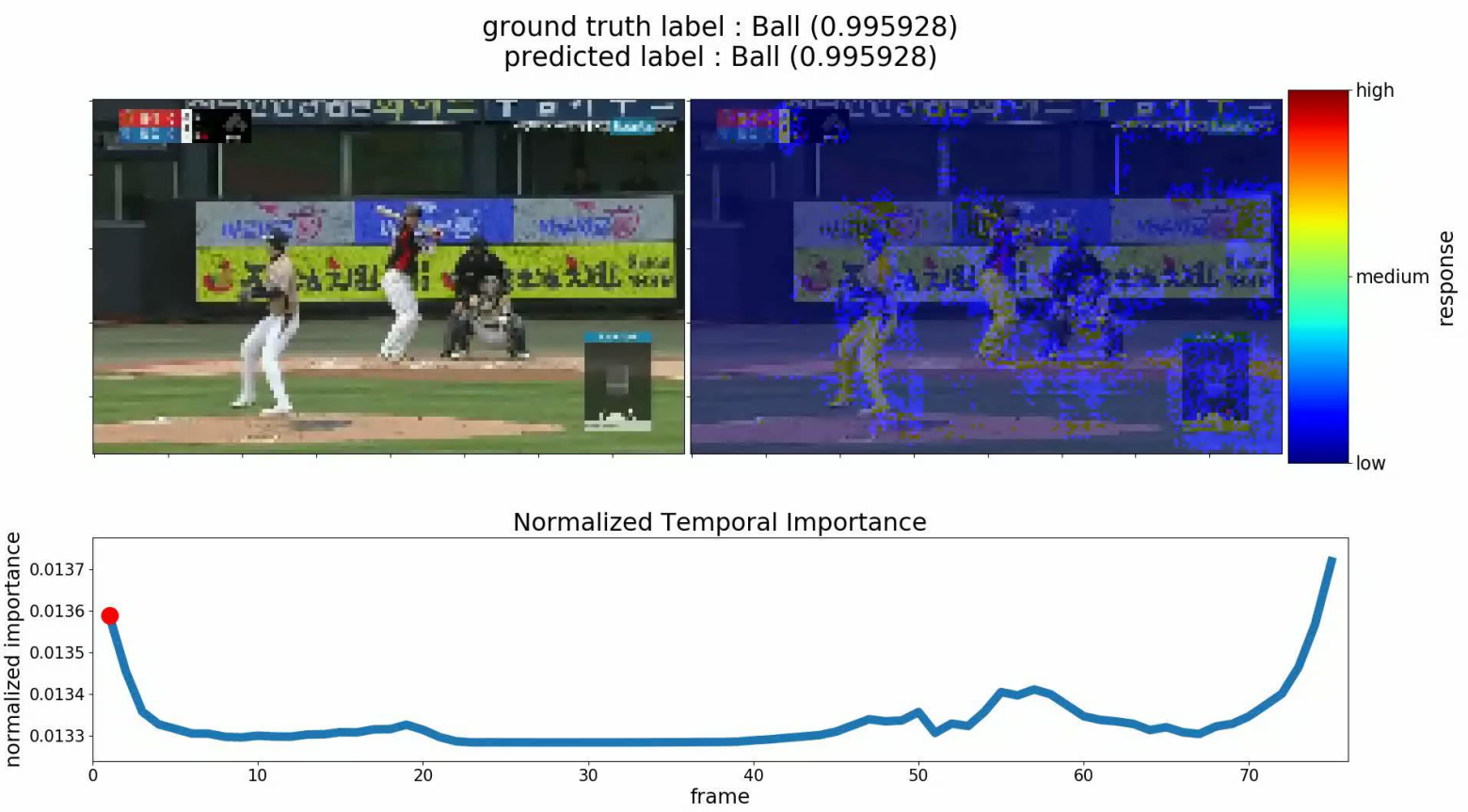
What is DNN looking at?



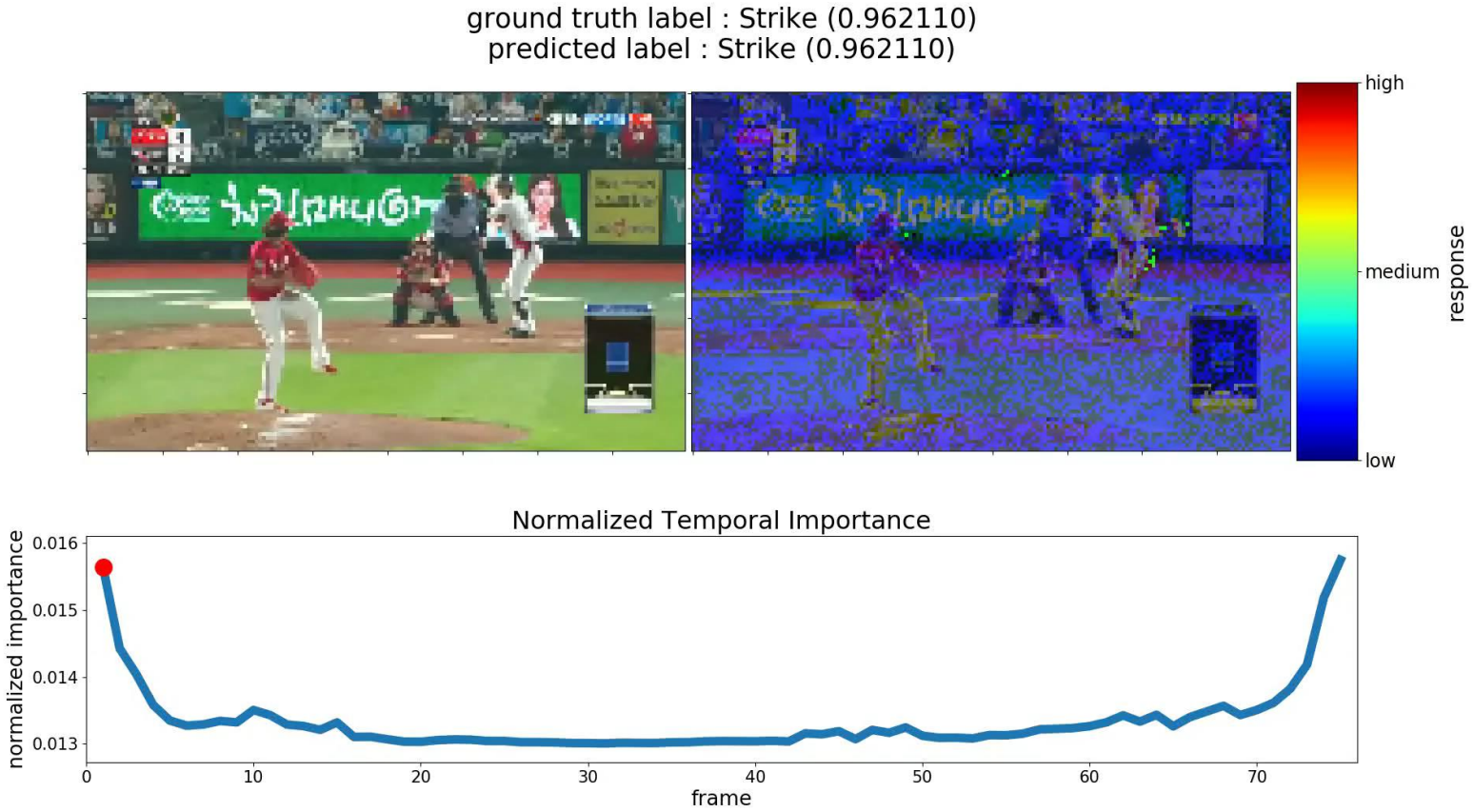


K. Simonyan et al., “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, ECCV 2014

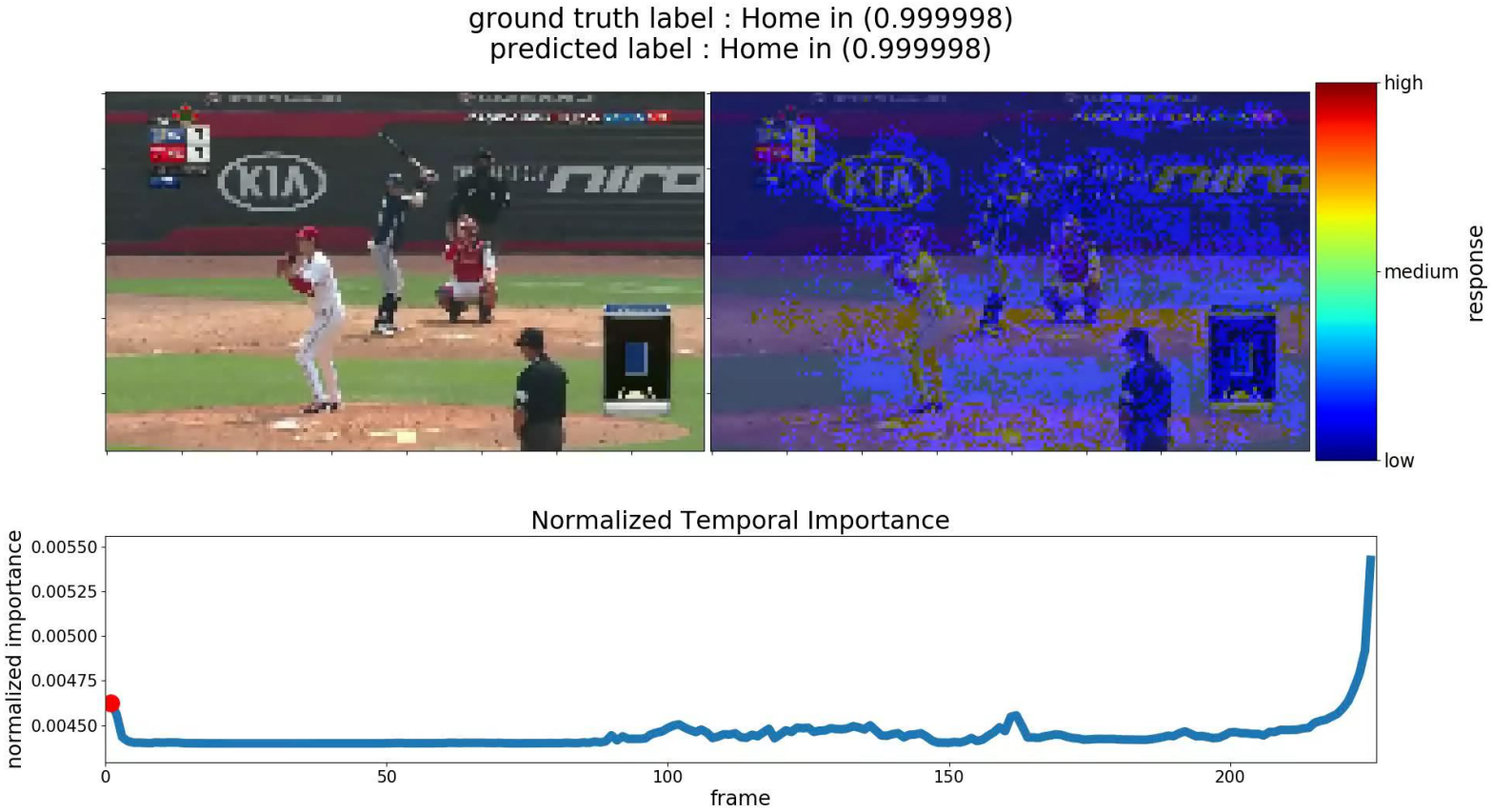
What is DNN looking at?



What is DNN looking at?



What is DNN looking at?



Summary

- Large database labelled semi-automatically
- Good for fine-grained motion recognition
- Can work on many tasks:

  - video recognition

  - video temporal localization

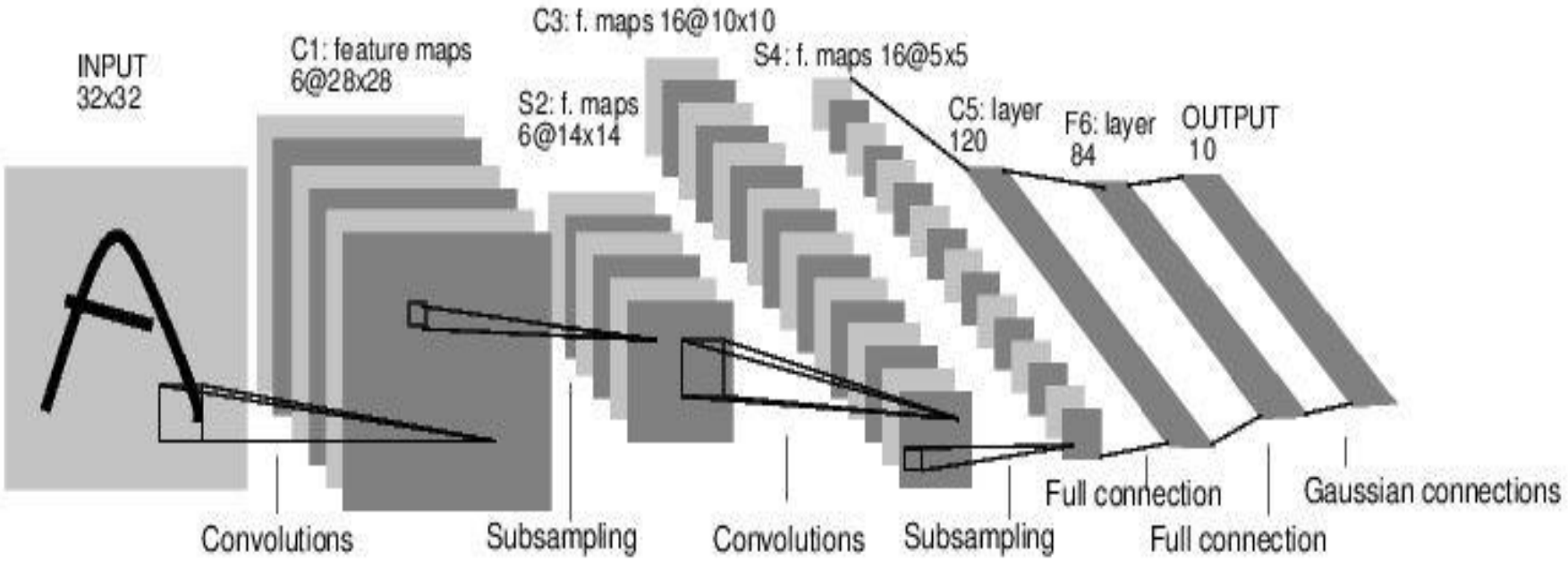
  - highlight generation

  - data imbalance problem

  - may be even more…

Deep Learning for Videos

ConvNets for images

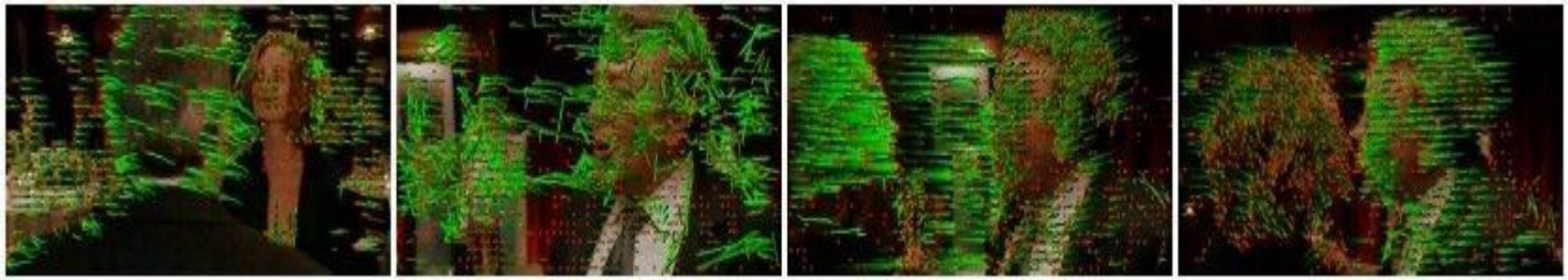


Feature-based approaches to Activity Recognition

Dense trajectories and motion boundary descriptors for action recognition
Wang et al., 2013

Action Recognition with Improved Trajectories
Wang and Schmid, 2013

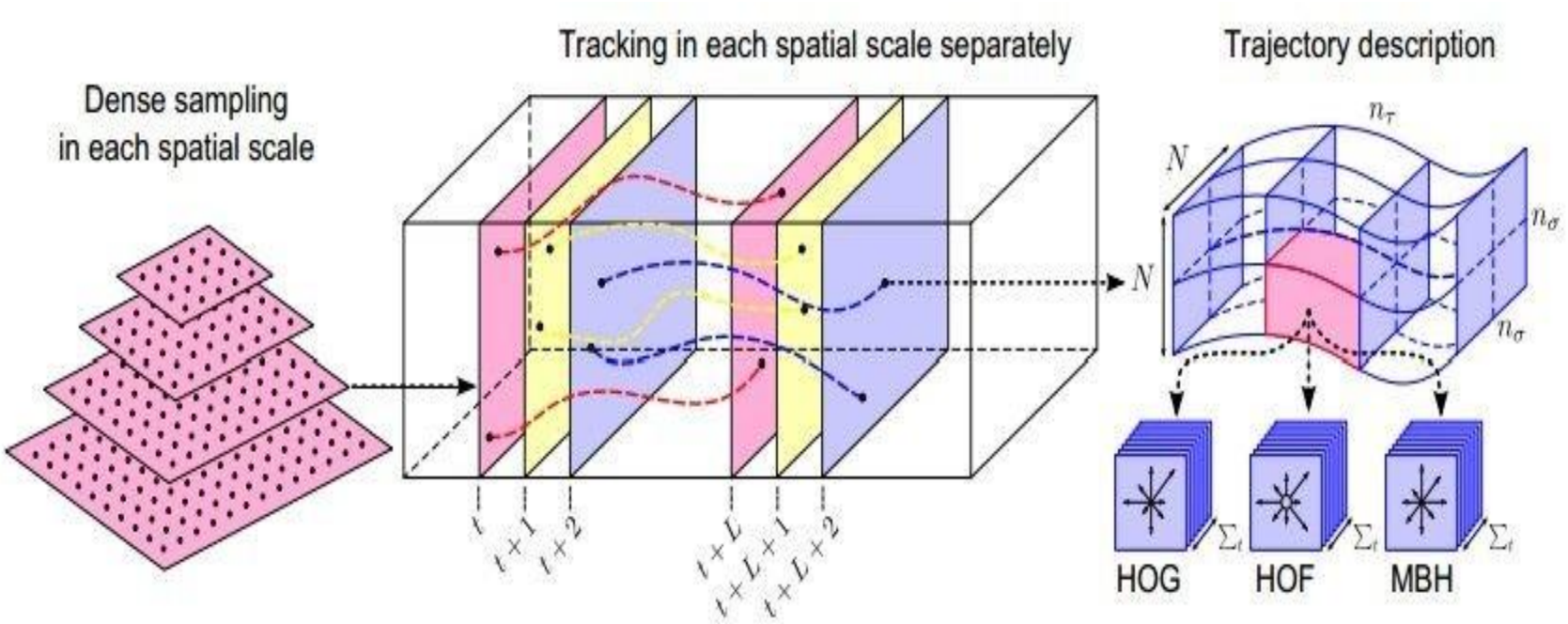
(code available!)



Dense trajectories

Dense trajectories and motion boundary descriptors for action recognition

Wang et al., 2013



detect feature points

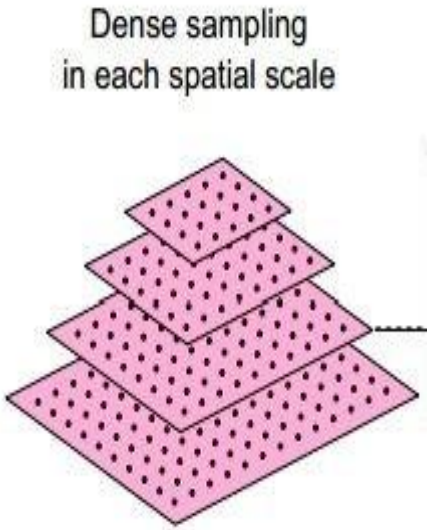
track features with optical flow

extract HOG/HOF/MBH features in the (stabilized) coordinate system of each tracklet

Dense trajectories and motion boundary descriptors for action recognition

Wang et al., 2013

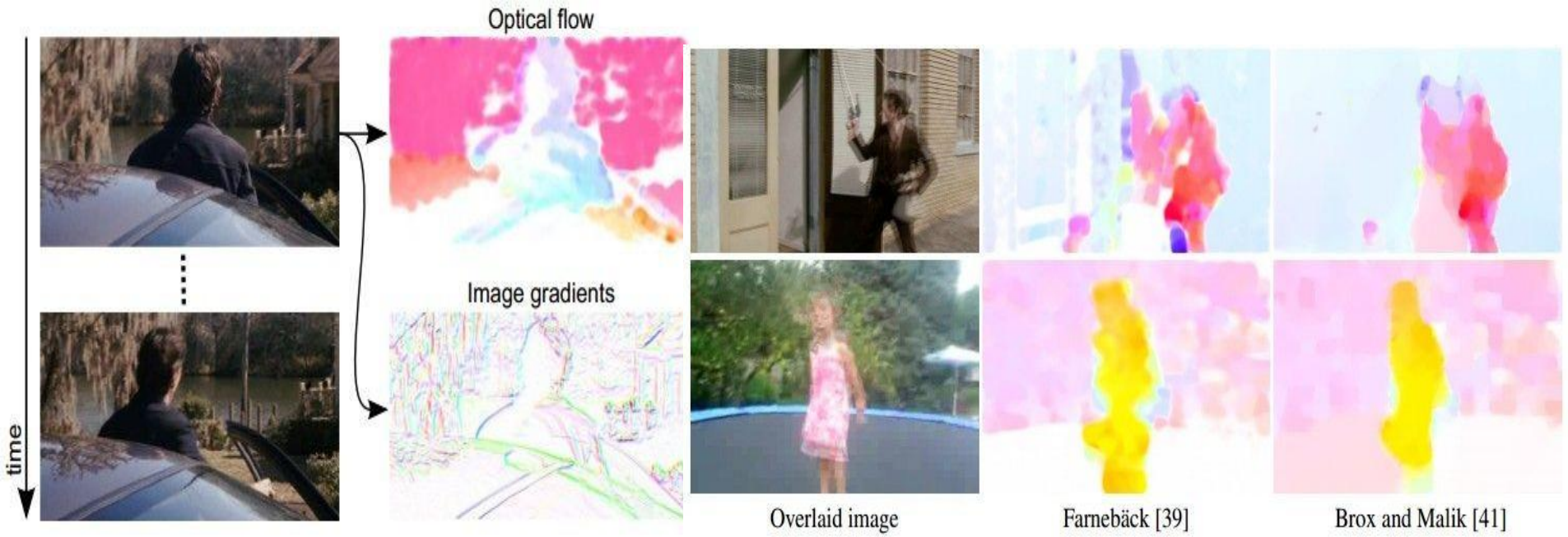
detected feature points



[J. Shi and C. Tomasi, “Good features to track,” CVPR 1994]
[Ivan Laptev 2005]

Dense trajectories and motion boundary descriptors for action recognition

Wang et al., 2013

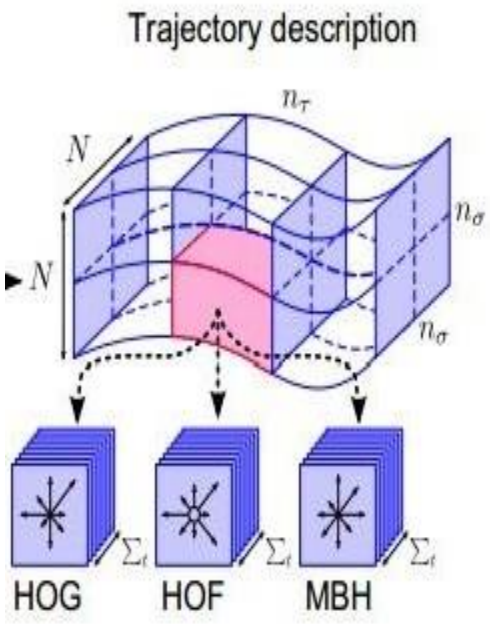


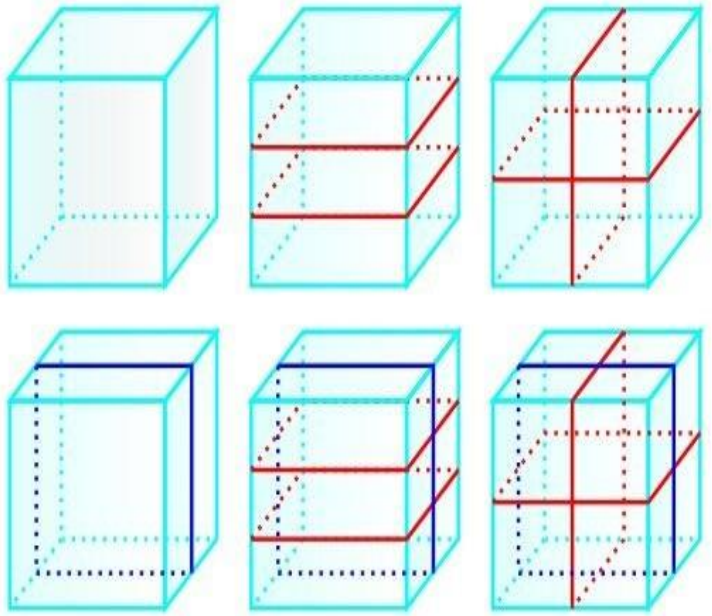
track each keypoint using **optical flow.**

[G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” 2003]
[T. Brox and J. Malik, “Large displacement optical flow: Descriptor matching in variational motion estimation,” 2011]

Dense trajectories and motion boundary descriptors for action recognition

Wang et al., 2013



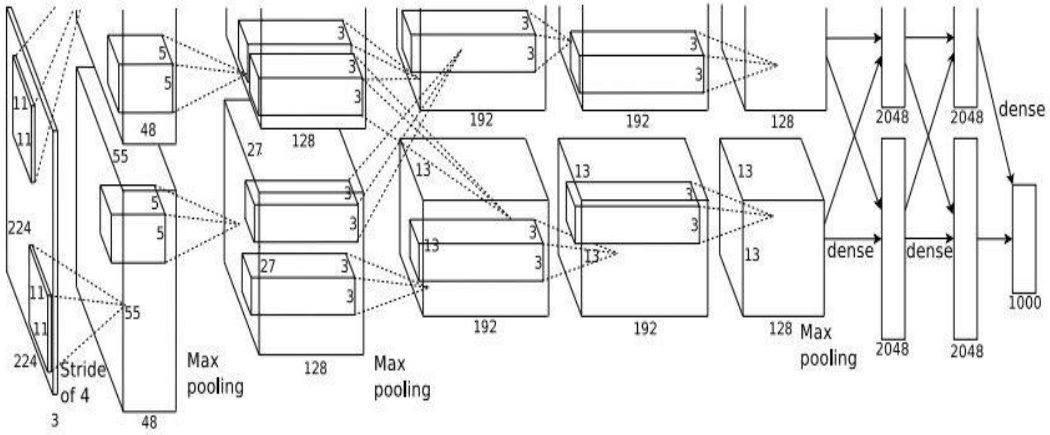


Extract features in the local coordinate system of each tracklet.

Accumulate into histograms, separately acco rding to multiple spatio-temporal layouts.

Case Study: AlexNet

[Krizhevsky et al. 2012]

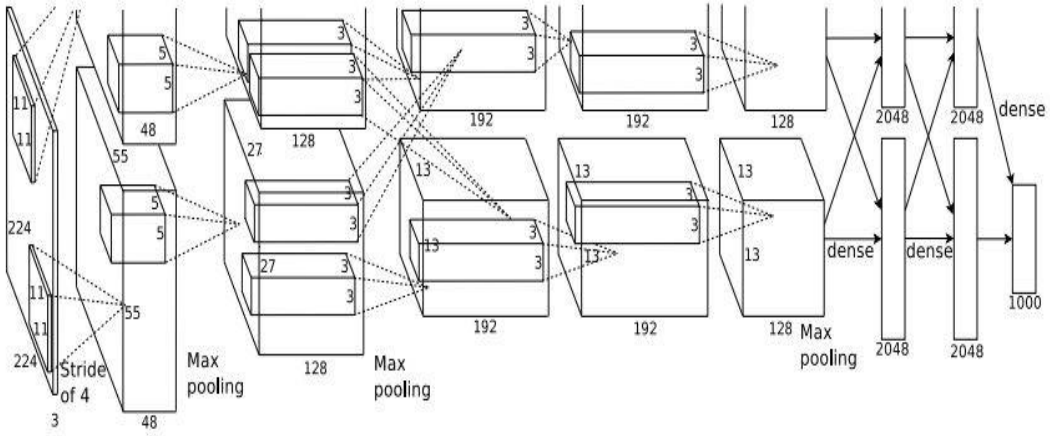


Input: 227x227x3 images

First layer (CONV1): 96 11x11 filters applied at stride 4
=>
Output volume **[55x55x96]**
Q: What if the input is now a small chunk of video? E.g. [227x227x3x15] ?

Case Study: AlexNet

[Krizhevsky et al. 2012]



Input: 227x227x3 images

First layer (CONV1): 96 11x11 filters applied at stride 4
=>
Output volume **[55x55x96]**
Q: What if the input is now a small chunk of video? E.g. [227x227x3x15] ?
A: Extend the convolutional filters in time, perform spatio-temporal convolutions!
E.g. can have 11x11xT filters, where T = 2..15.

Spatio-Temporal ConvNets

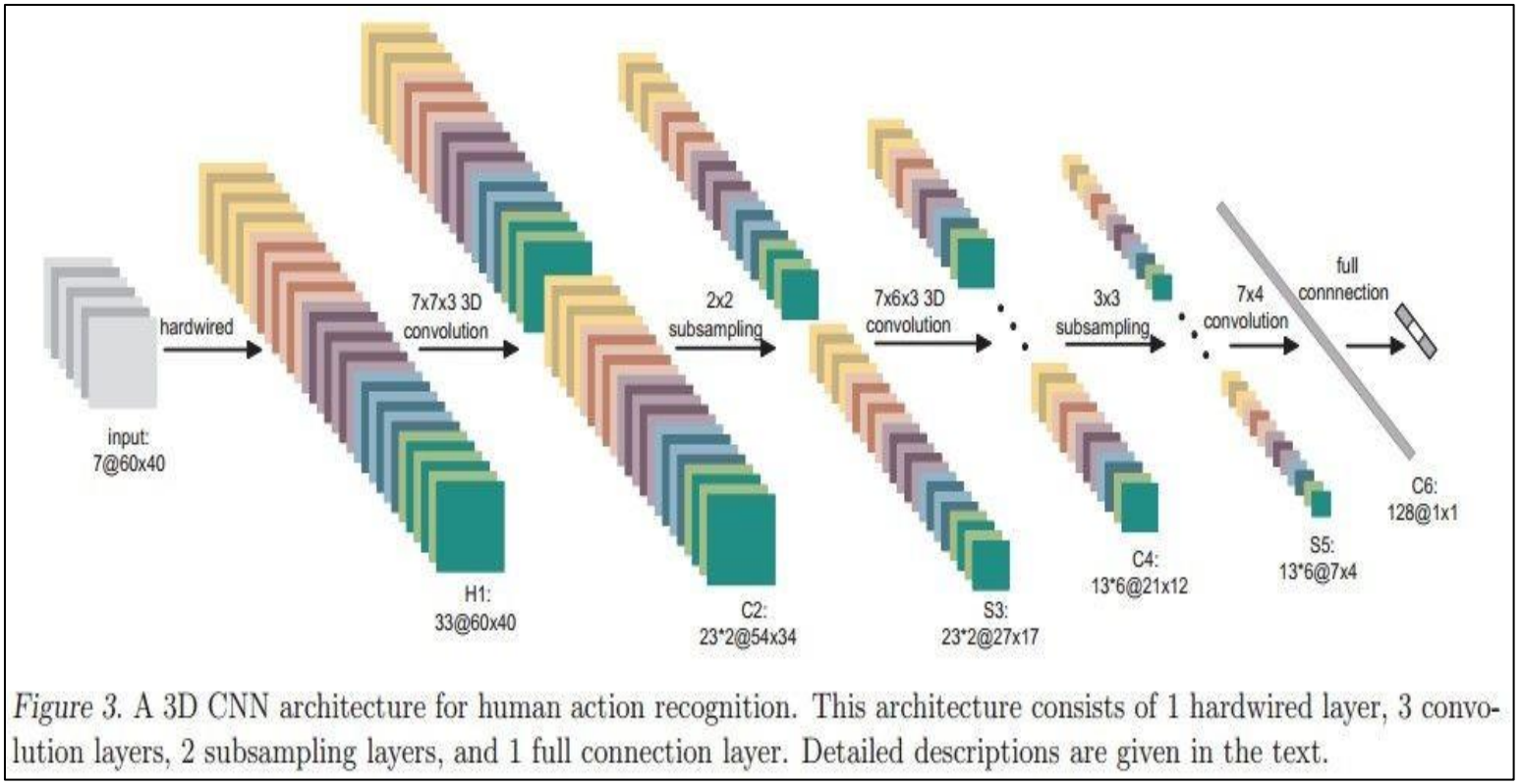
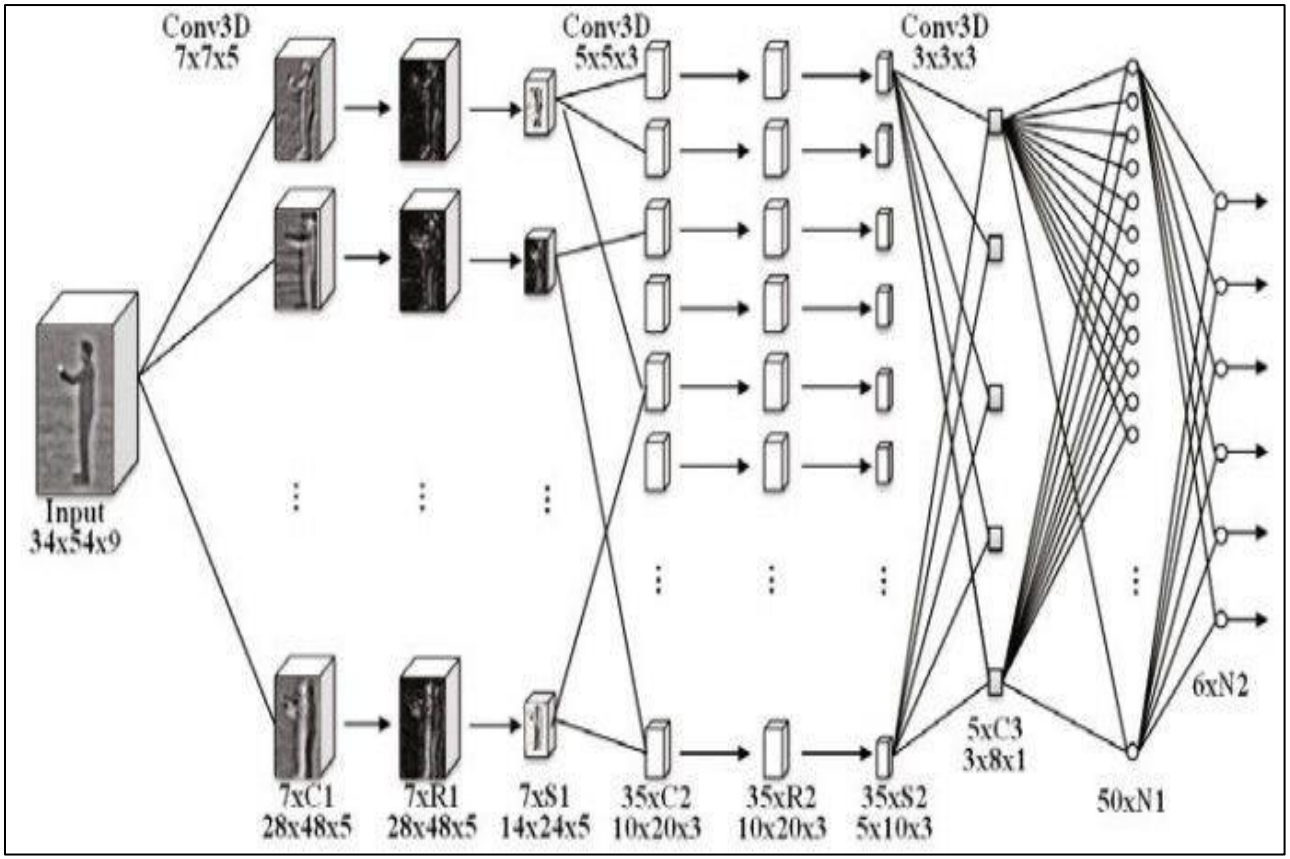


Figure 3. A 3D CNN architecture for human action recognition. This architecture consists of 1 hardwired layer, 3 convolution layers, 2 subsampling layers, and 1 full connection layer. Detailed descriptions are given in the text.

[3D Convolutional Neural Networks for Human Action Recognition, Ji et al., 2010]

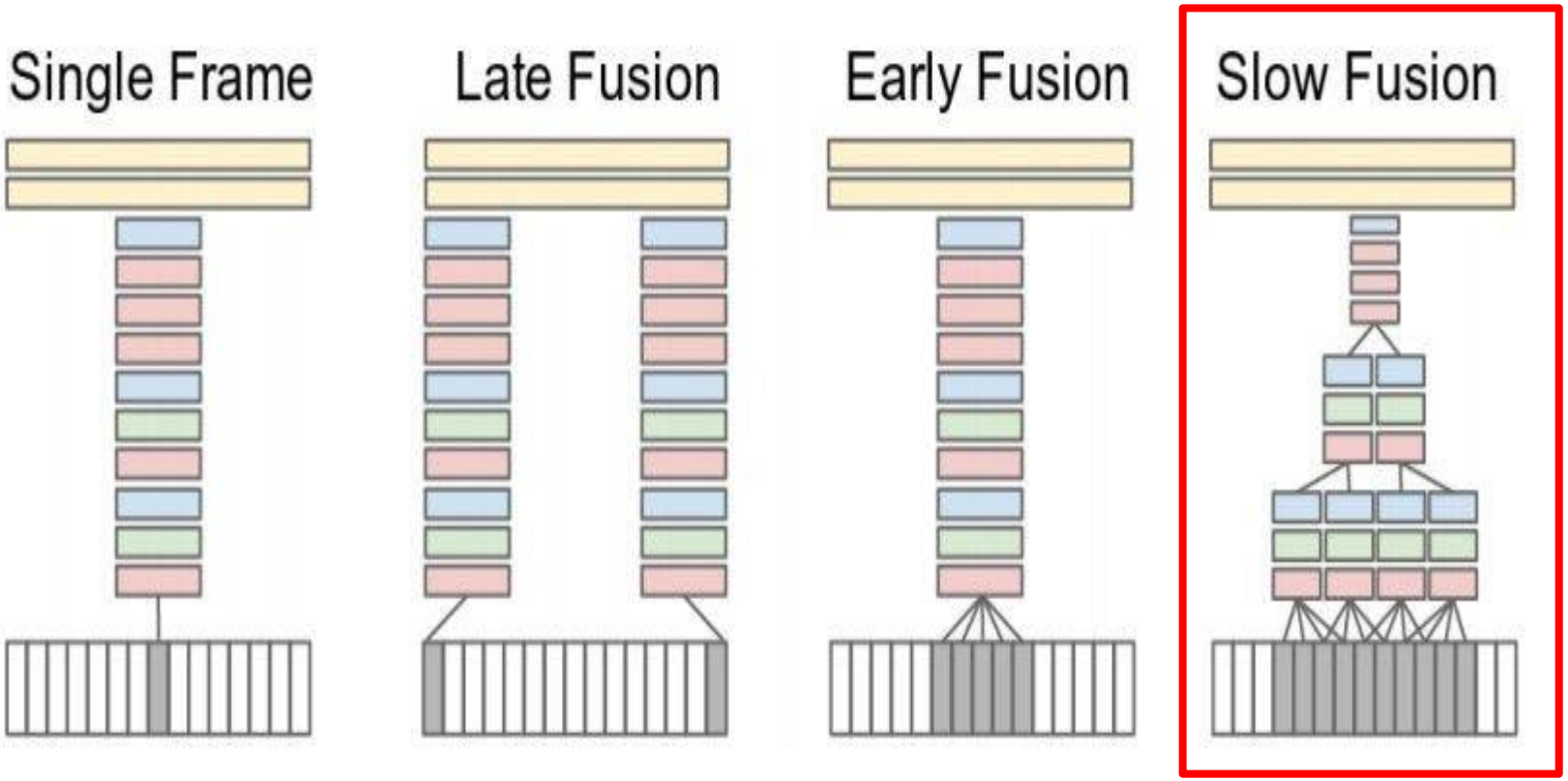
Spatio-Temporal ConvNets



Sequential Deep Learning for Human Action Recognition, Baccouche et al., 2011

Spatio-Temporal ConvNets

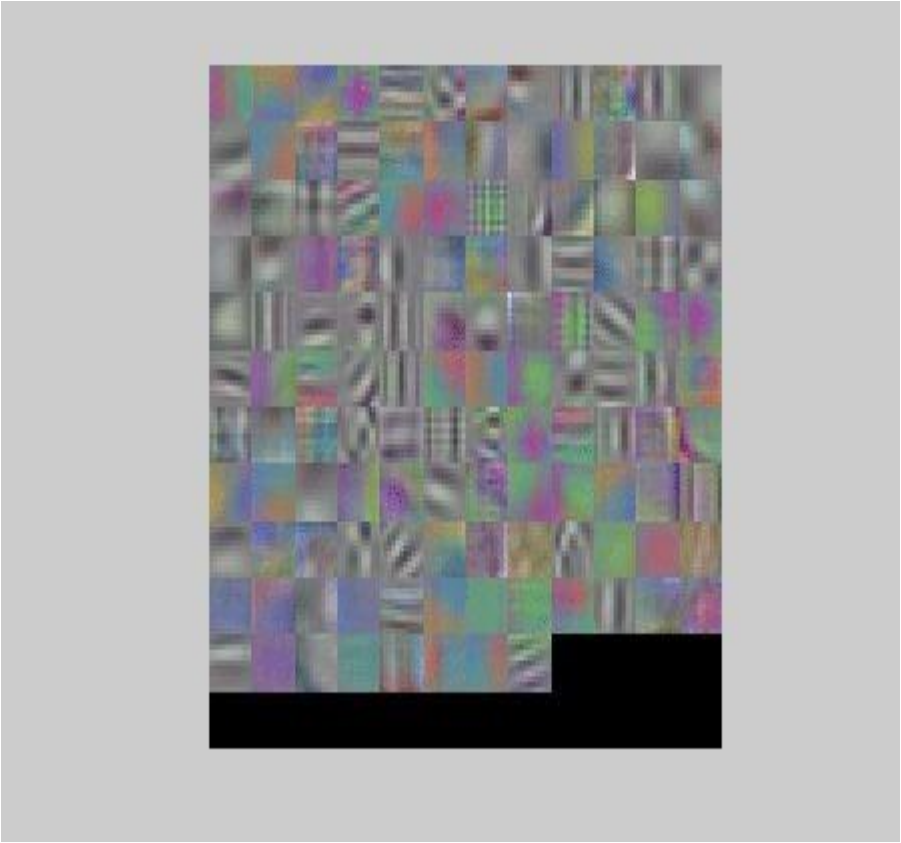
spatio-temporal convolutions; worked best.



[Large-scale Video Classification with Convolutional Neural Networks, Karpathy et al., 2014]

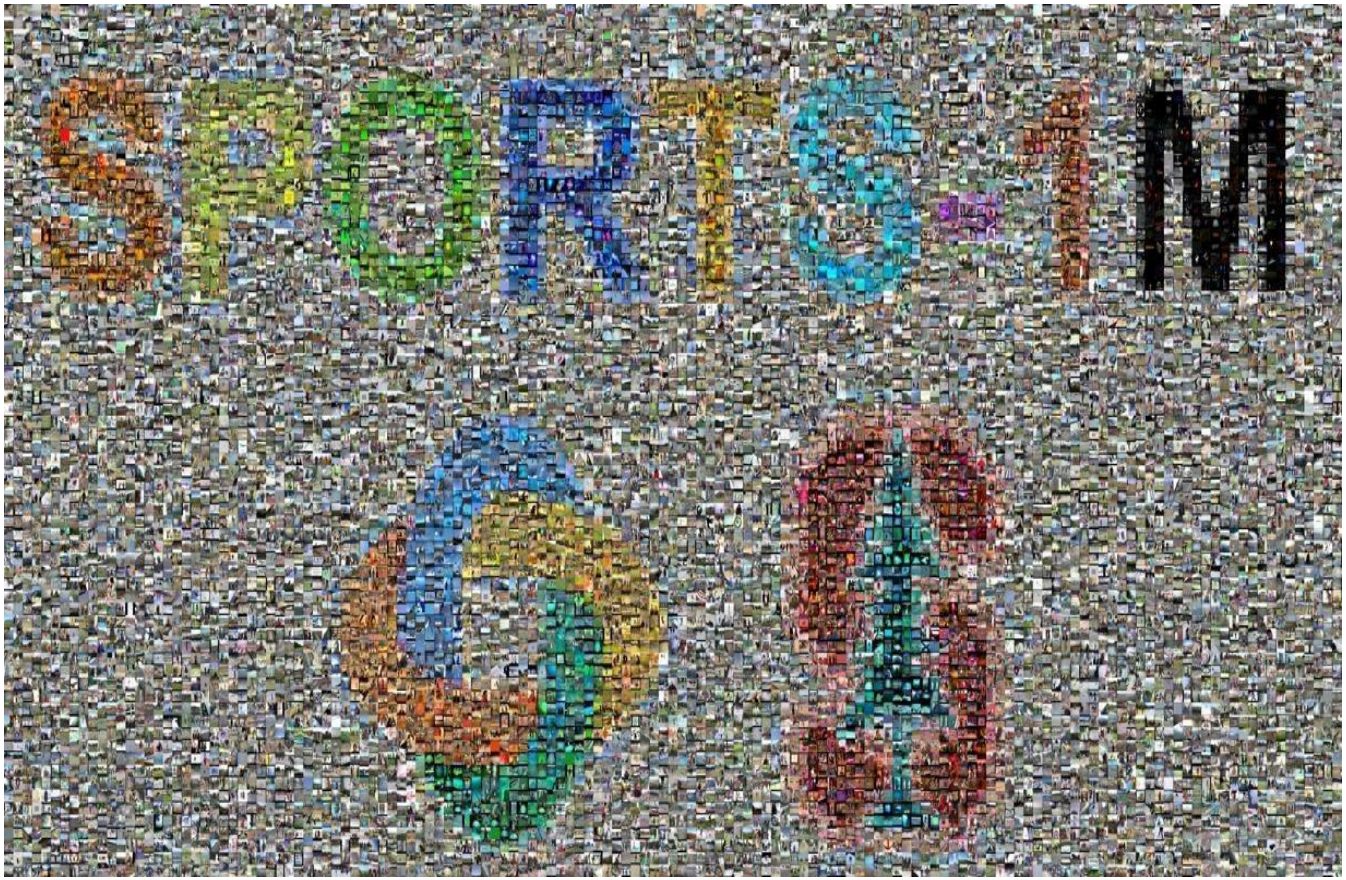
Spatio-Temporal ConvNets

Learned filters on the first layer



[Large-scale Video Classification with Convolutional Neural Networks, Karpathy et al., 2014]

Spatio-Temporal ConvNets



1 million videos
487 sports classes

[Large-scale Video Classification with Convolutional Neural Networks, Karpathy et al., 2014]

Spatio-Temporal ConvNets

Model	Clip Hit@1	Video Hit@1	Video Hit@5
Feature Histograms + Neural Net	-	55.3	-
Single-Frame	41.1	59.3	77.7
Single-Frame + Multires	**42.4**	**60.0**	**78.5**
Single-Frame Fovea Only	30.0	49.9	72.8
Single-Frame Context Only	38.1	56.0	77.2
Early Fusion	38.9	57.7	76.8
Late Fusion	40.7	59.3	78.7
Slow Fusion	**41.9**	60.9	**80.2**
CNN Average (Single+Early+Late+Slow)	41.4	63.9	82.4

The motion information didn't add all that much...

[Large-scale Video Classification with Convolutional Neural Networks, Karpathy et al., 2014]

Spatio-Temporal ConvNets

Spatio-Temporal ConvNets

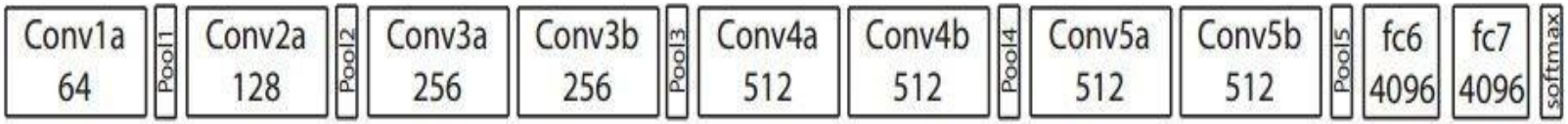
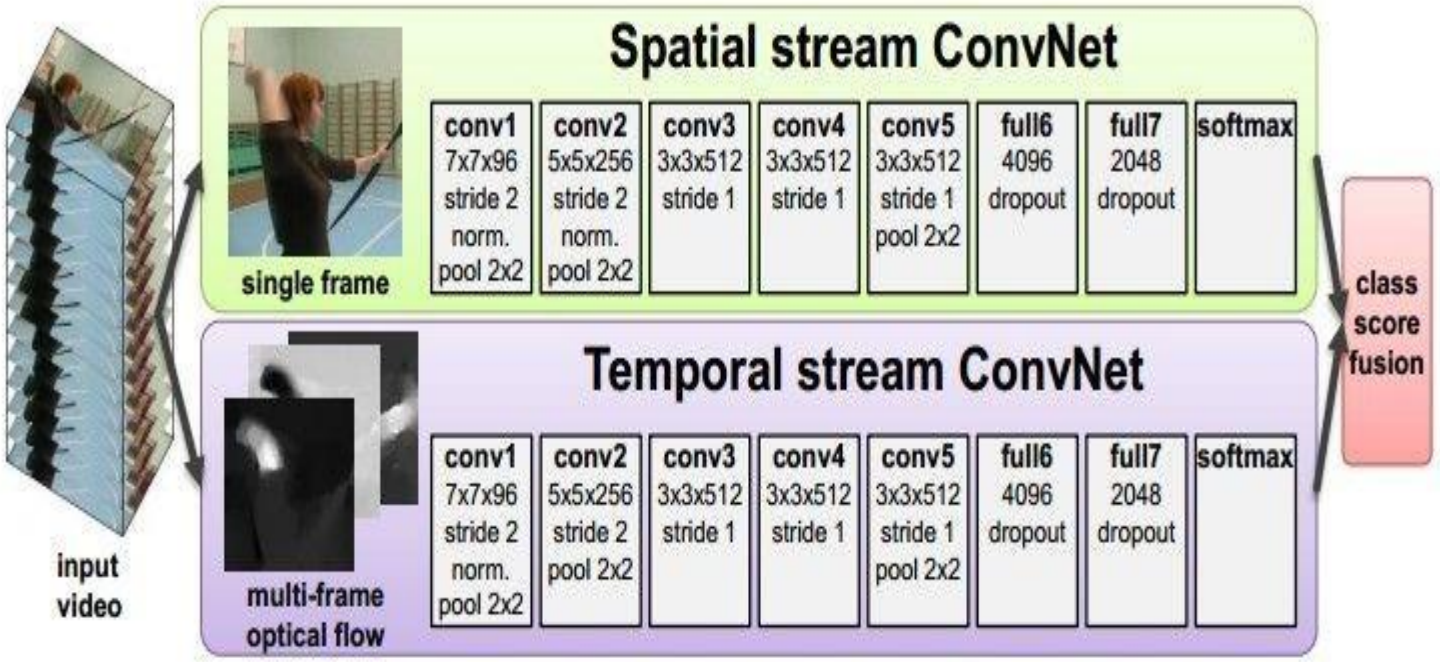


Figure 3. **C3D architecture**. C3D net has 8 convolution, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. All 3D convolution kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. Number of filters are denoted in each box. The 3D pooling layers are denoted from `pool1` to `pool5`. All pooling kernels are $2 \times 2 \times 2$, except for `pool1` is $1 \times 2 \times 2$. Each fully connected layer has 4096 output units.

3D VGGNet, basically.

[Learning Spatiotemporal Features with 3D Convolutional Networks, Tran et al. 2015]

Spatio-Temporal ConvNets

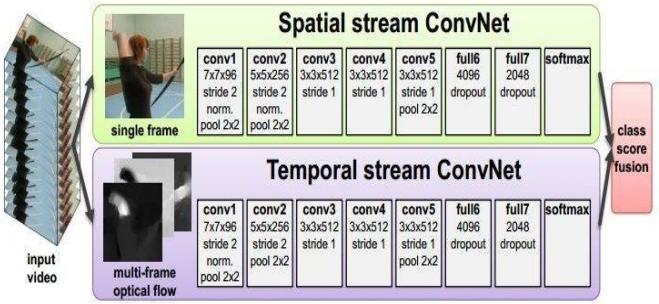


(of VGGNet fame)

[Two-Stream Convolutional Networks for Action Recognition in Videos, **Simonyan** and Zisserman 2014]

[T. Brox and J. Malik, “Large displacement optical flow: Descriptor matching in variational motion estimation,” 2011]

Spatio-Temporal ConvNets



Spatial stream ConvNet	73.0%	40.5%
Temporal stream ConvNet	83.7%	54.6%
Two-stream model (fusion by averaging)	86.9%	58.0%
Two-stream model (fusion by SVM)	**88.0%**	**59.4%**

Two-stream version works much better than either alone.

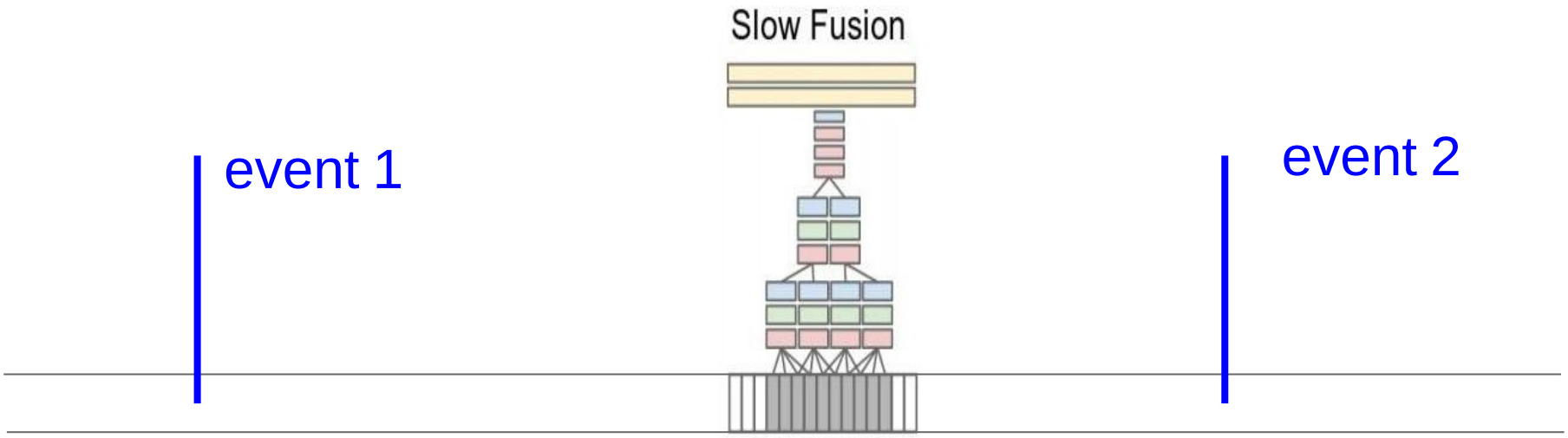
[Two-Stream Convolutional Networks for Action Recognition in Videos, **Simonyan** and Zisserman 2014]

[T. Brox and J. Malik, “Large displacement optical flow: Descriptor matching in variational motion estimation,” 2011]

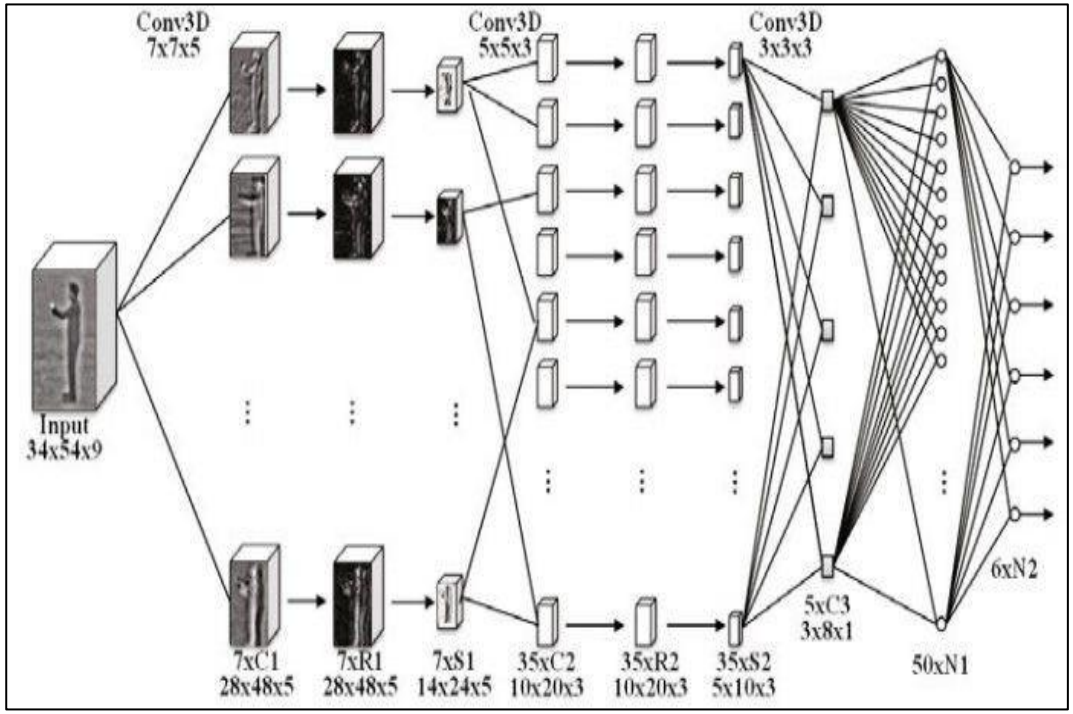
Long-time Spatio-Temporal ConvNets

All 3D ConvNets so far used local motion cues to get extra accuracy (e.g. half a second or so)

Q: what if the temporal dependencies of interest are much much longer? E.g. several seconds?



Long-time Spatio-Temporal ConvNets

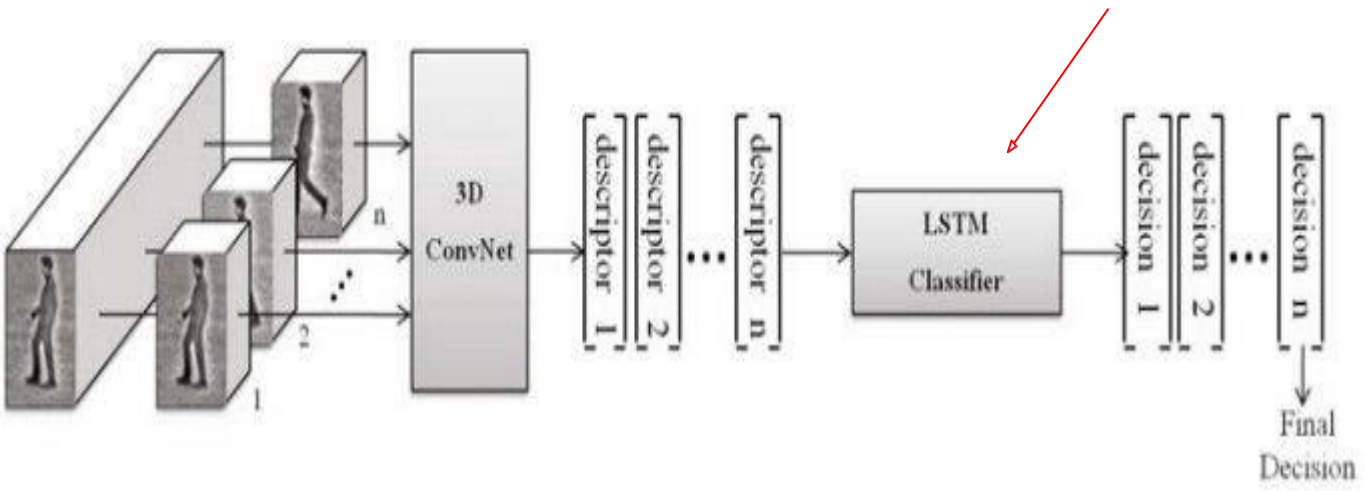


(This paper was way ahead of its time. Cited 65 times.)

Sequential Deep Learning for Human Action Recognition, Baccouche et al., **2011**

Long-time Spatio-Temporal ConvNets

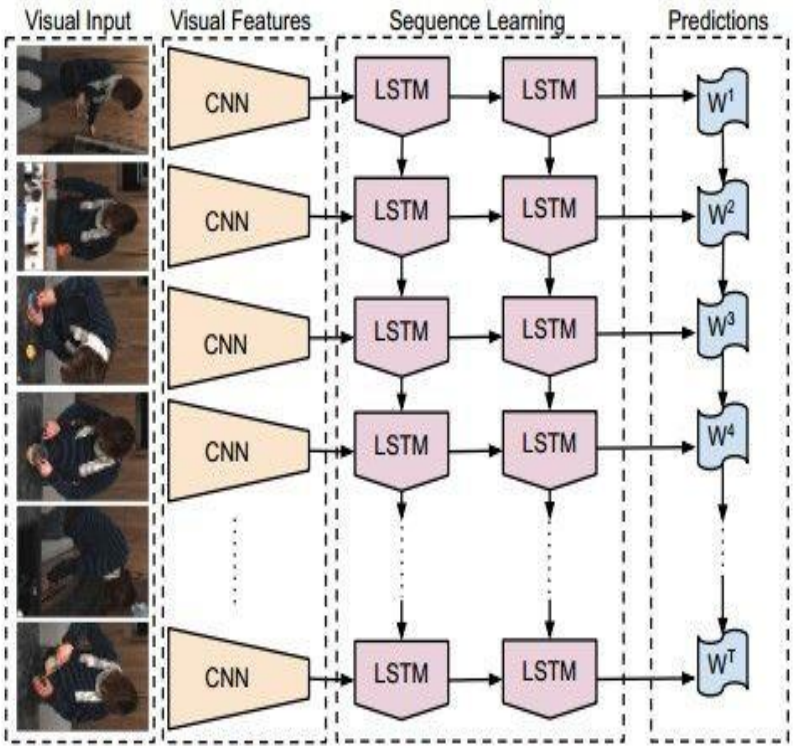
LSTM way before it was cool



(This paper was way ahead of its time. Cited 65 times.)

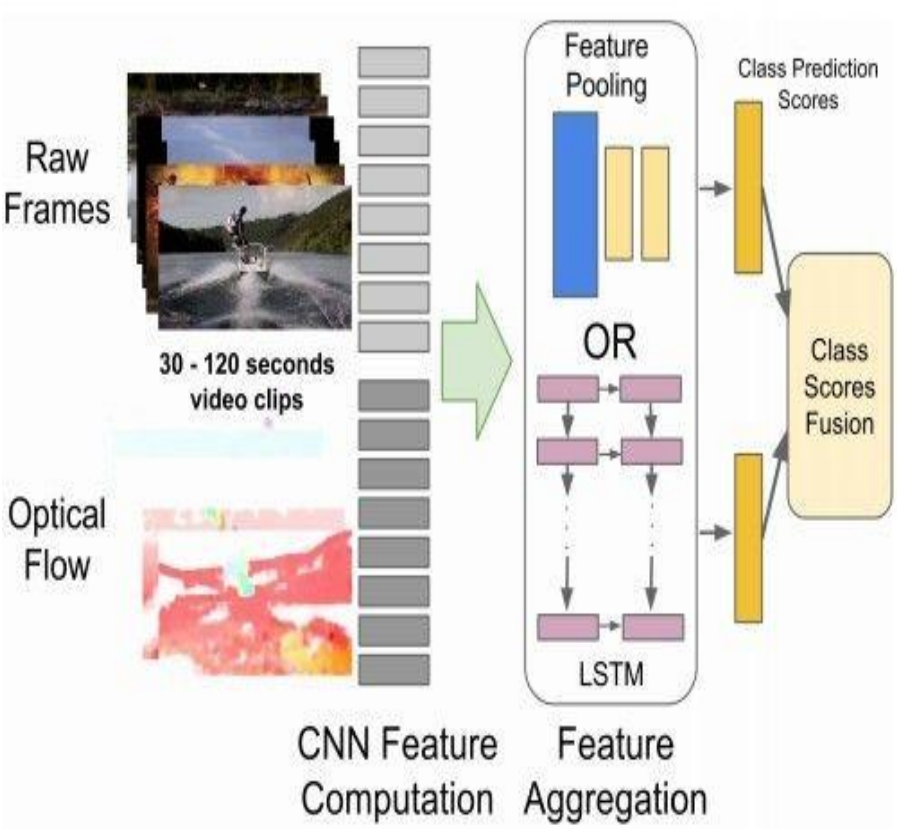
Sequential Deep Learning for Human Action Recognition, Baccouche et al., **2011**

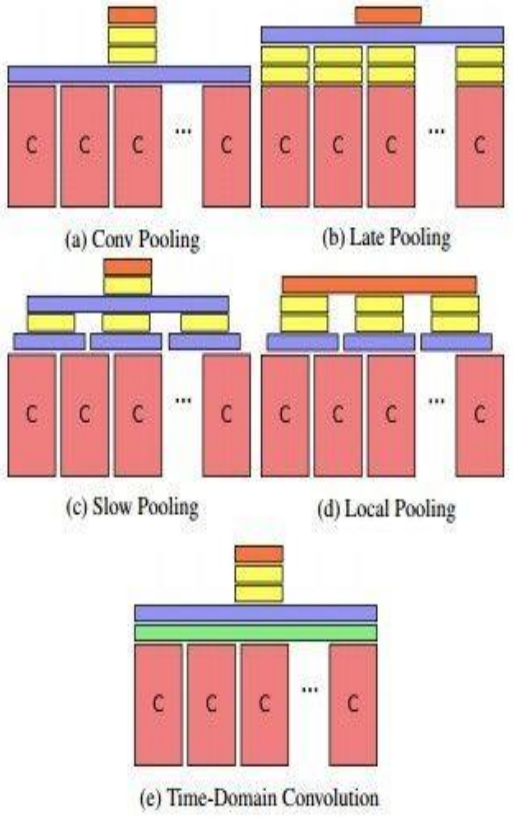
Long-time Spatio-Temporal ConvNets



[Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al., 2015]

Long-time Spatio-Temporal ConvNets





[Beyond Short Snippets: Deep Networks for Video Classification, Ng et al., 2015]

Summary so far

We looked at two types of architectural patterns:

1. Model temporal motion locally (3D CONV)

2. Model temporal motion globally (LSTM / RNN)

\+ Fusions of both approaches at the same time.

Summary so far

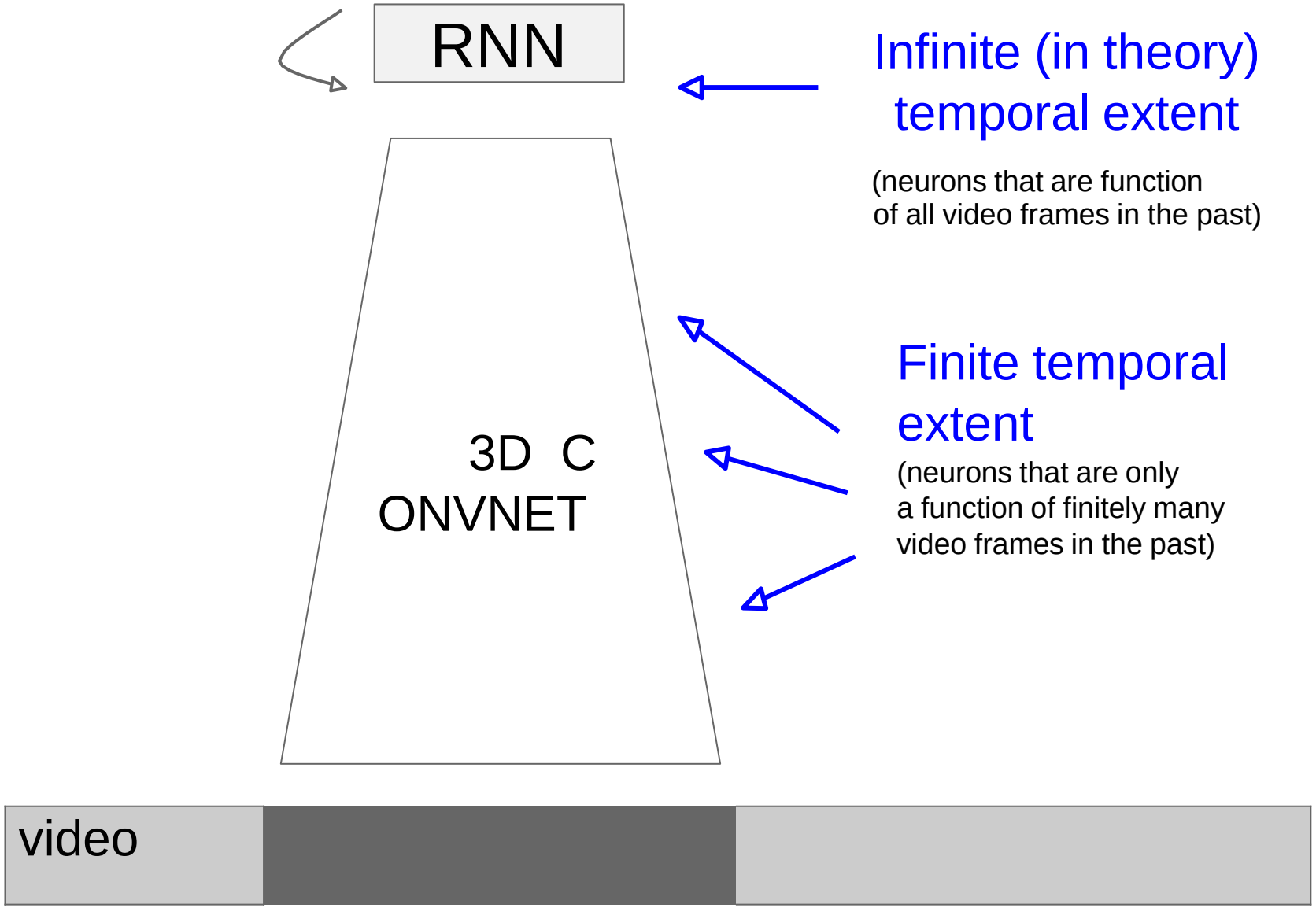
We looked at two types of architectural patterns:

1. Model temporal motion locally (3D CONV)

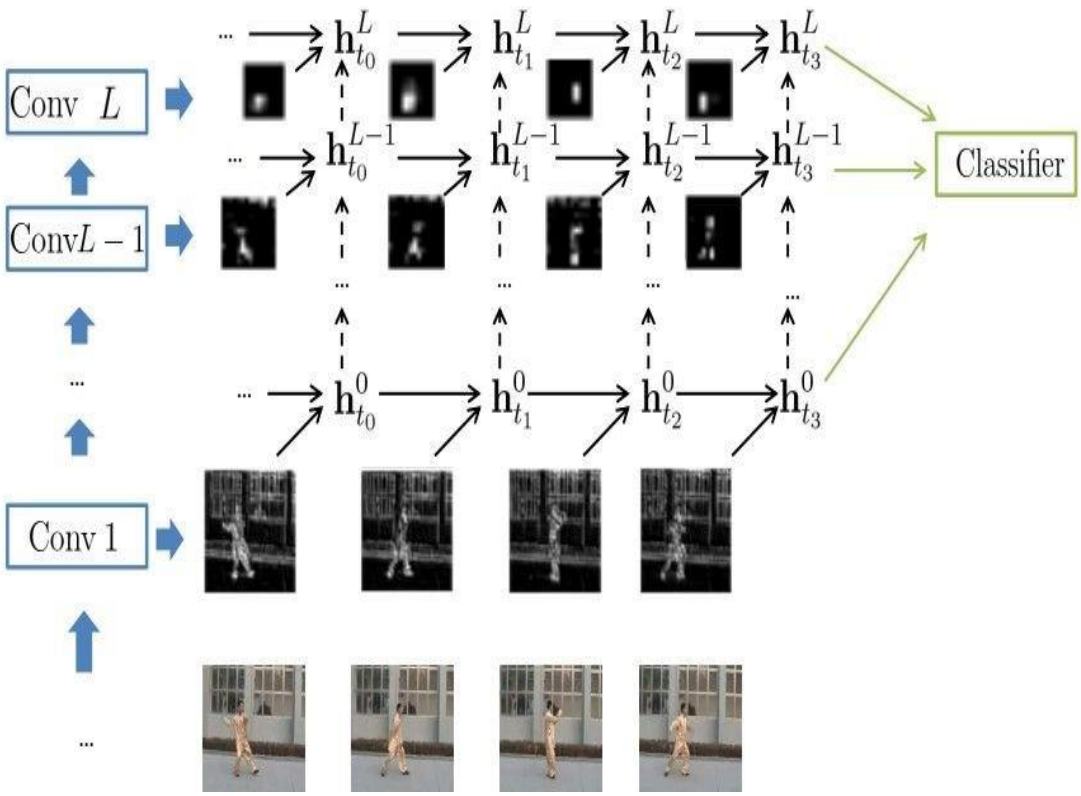
2. Model temporal motion globally (LSTM / RNN)

\+ Fusions of both approaches at the same time.

There is another (cleaner) way!


RNN
Infinite (in theory)
temporal extent
(neurons that are function
of all video frames in the past)
3D C
ONVNET
Finite temporal
extent
(neurons that are only
a function of finitely many
video frames in the past)
video

Long-time Spatio-Temporal ConvNets



Beautiful:
All neurons in the ConvNet are recurrent.

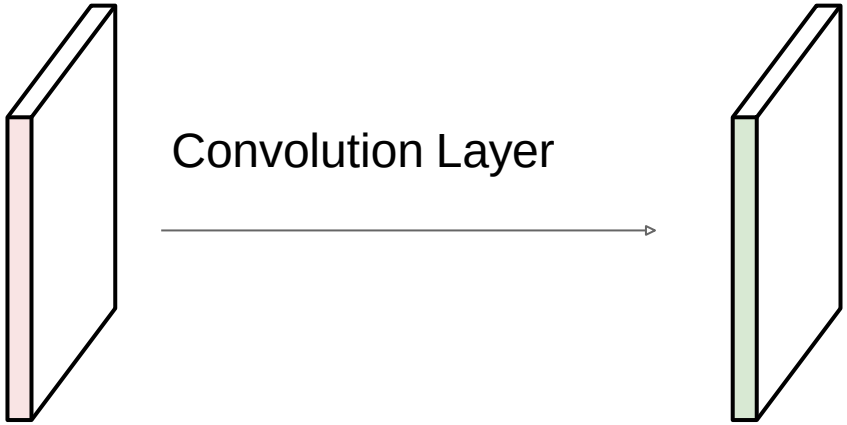
$$
\begin{aligned}
\mathbf{z}_t^l &= \sigma(\mathbf{W}_z^l * \mathbf{x}_t^l + \mathbf{U}_z^l * \mathbf{h}_{t-1}^l), \\
\mathbf{r}_t^l &= \sigma(\mathbf{W}_r^l * \mathbf{x}_t^l + \mathbf{U}_r^l * \mathbf{h}_{t-1}^l), \\
\tilde{\mathbf{h}}_t^l &= \tanh(\mathbf{W}^l * \mathbf{x}_t^l + \mathbf{U} * (\mathbf{r}_t^l \odot \mathbf{h}_{t-1}^l), \\
\mathbf{h}_t^l &= (1 - \mathbf{z}_t^l)\mathbf{h}_{t-1}^l + \mathbf{z}_t^l \tilde{\mathbf{h}}_t^l,
\end{aligned}
$$

Only requires (existing) 2D CONV routines. No need for 3D spatio-temporal CONV.

[Delving Deeper into Convolutional Networks for Learning Video Representations, Ballas et al., 2016]

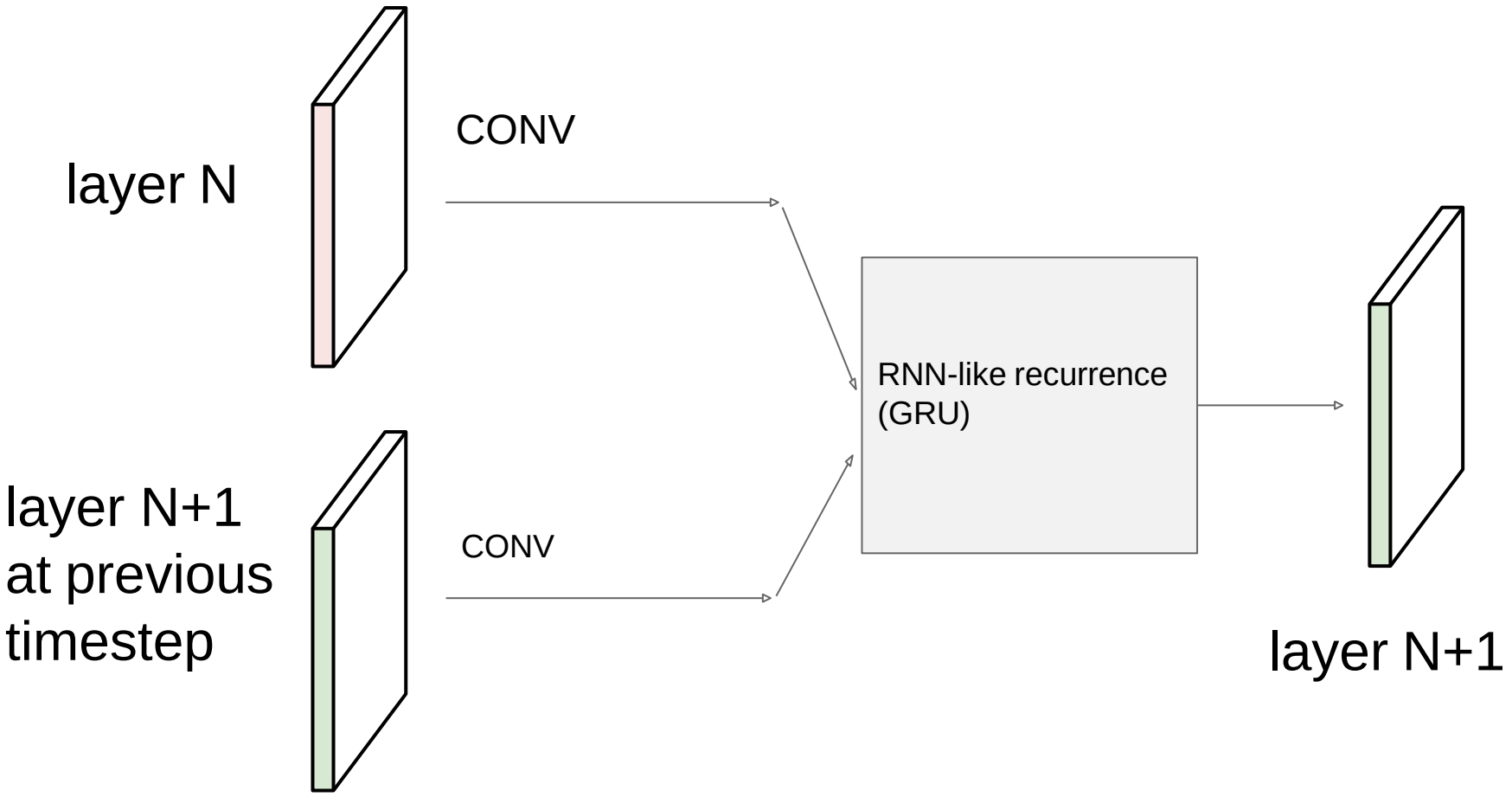
Long-time Spatio-Temporal ConvNets

Normal ConvNet:



[Delving Deeper into Convolutional Networks for Learning Video Representations, Ballas et al., 2016]

Long-time Spatio-Temporal ConvNets



[Delving Deeper into Convolutional Networks for Learning Video Representations, Ballas et al., 2016]

Long-time Spatio-Temporal ConvNets

Recall: RNNs

$$h_t = f_W(h_{t-1}, x_t)$$

Vanilla RNN

$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$

GRU

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1}), \\
\mathbf{r}_t &= \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1}), \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W} \mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t \tilde{\mathbf{h}}_t,
\end{aligned}$$

LSTM

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

$$c_t^l = f \odot c_{t-1}^l + i \odot g$$

$$h_t^l = o \odot \tanh(c_t^l)$$

[Delving Deeper into Convolutional Networks for Learning Video Representations, Ballas et al., 2016]

Long-time Spatio-Temporal ConvNets

Recall: RNNs

$$h_t = f_W(h_{t-1}, x_t)$$

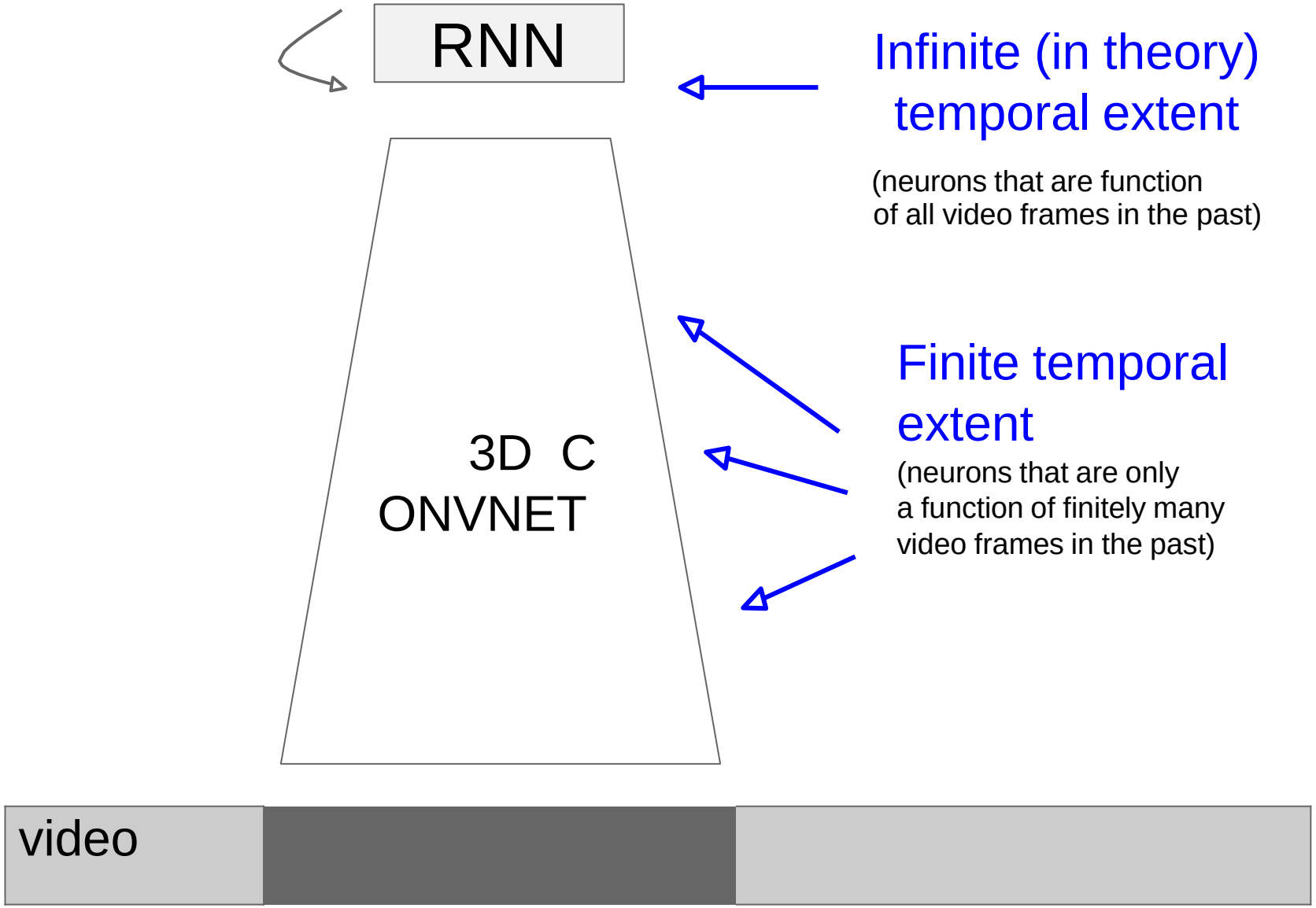
GRU

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1}), \\
\mathbf{r}_t &= \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1}), \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}\mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t \tilde{\mathbf{h}}_t,
\end{aligned}$$

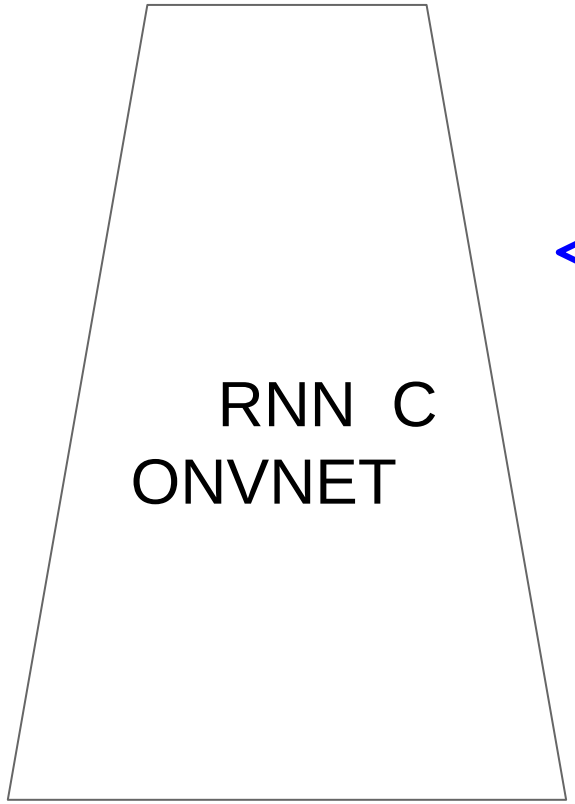
Matrix multiply
=> C
ONV

$$\begin{aligned}
\mathbf{z}_t^l &= \sigma(\mathbf{W}_z^l * \mathbf{x}_t^l + \mathbf{U}_z^l * \mathbf{h}_{t-1}^l), \\
\mathbf{r}_t^l &= \sigma(\mathbf{W}_r^l * \mathbf{x}_t^l + \mathbf{U}_r^l * \mathbf{h}_{t-1}^l), \\
\tilde{\mathbf{h}}_t^l &= \tanh(\mathbf{W}^l * \mathbf{x}_t^l + \mathbf{U} * (\mathbf{r}_t^l \odot \mathbf{h}_{t-1}^l), \\
\mathbf{h}_t^l &= (1 - \mathbf{z}_t^l)\mathbf{h}_{t-1}^l + \mathbf{z}_t^l \tilde{\mathbf{h}}_t^l,
\end{aligned}$$

[Delving Deeper into Convolutional Networks for Learning Video Representations, Ballas et al., 2016]


RNN
Infinite (in theory)
temporal extent
(neurons that are function
of all video frames in the past)
3D C
ONVNET
Finite temporal
extent
(neurons that are only
a function of finitely many
video frames in the past)
video

i.e. we obtain:



Infinite (in theory)
temporal extent

(neurons that are function
of all video frames in the past)

Summary

- You think you need a Spatio-Temporal Fancy Video ConvNet
- STOP. Do you really?
- Okay fine: do you want to model:
  - local motion? (use 3D CONV), or
  - global motion? (use LSTM).
- Try out using Optical Flow in a second stream (can work better sometimes)
- Try out GRU-RCN! (imo best model)

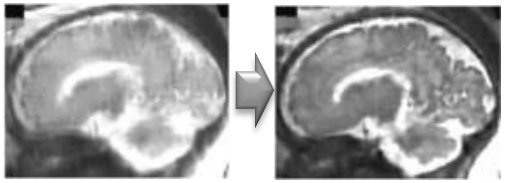
Deep Video Super-Resolution Network Using Dynamic Upsampling Filters Without Explicit Motion Compensation

Younghyun Jo Seoung Wug Oh Jaeyeon Kang Seon Joo Kim

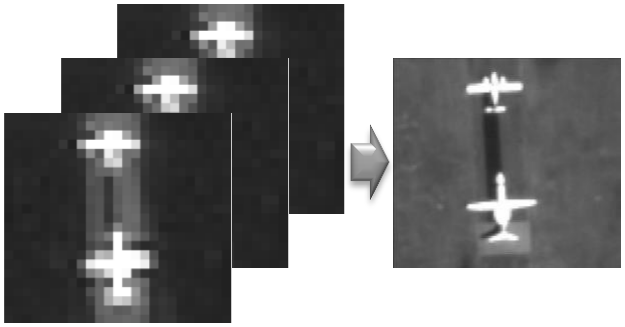
Yonsei University

CVPR18

Super-Resolution (SR)



Medical imaging



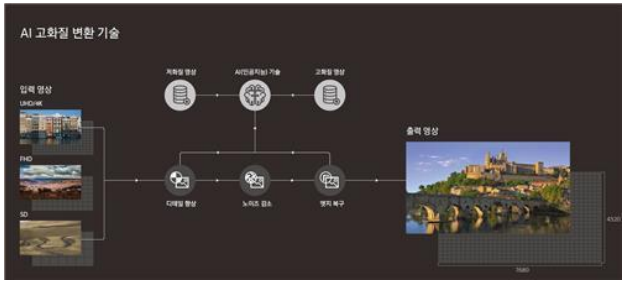
Satellite imaging

Surveillance camera

8K UHD TV

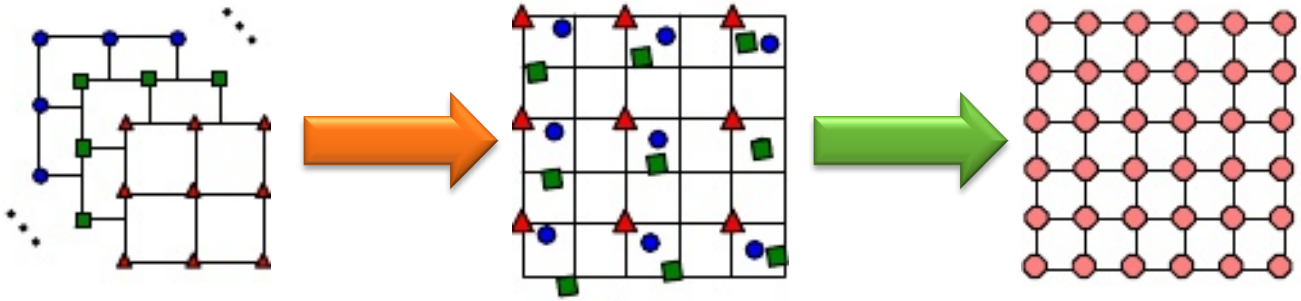
[CES 2018] 삼성전자, 85인치 QLED TV 공개…"AI 기술로 저화질→고화질"

2018. 01. 07.



Video SR

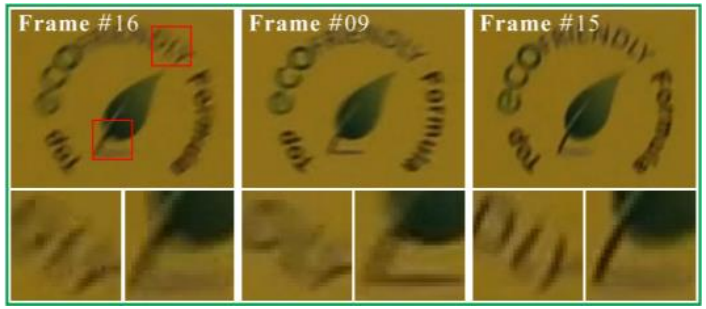
- Conventional video SR algorithm consists of two steps.
  - Motion compensation (MC) followed by restoration process.



Set of low-res (LR) frames | Motion compensated | Reconstructed high-res (HR) frame

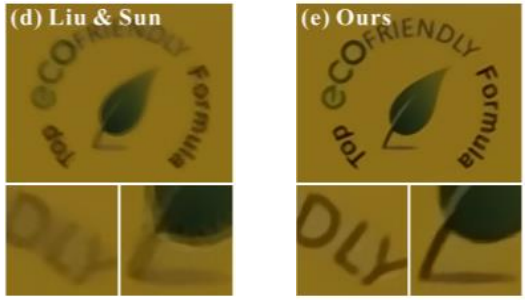
- But it has **a problem**.
  - SR results rely heavily on the accurate motion estimation.

Input LR frames



Video SR

The result of inaccurate(d) and accurate MC(e)

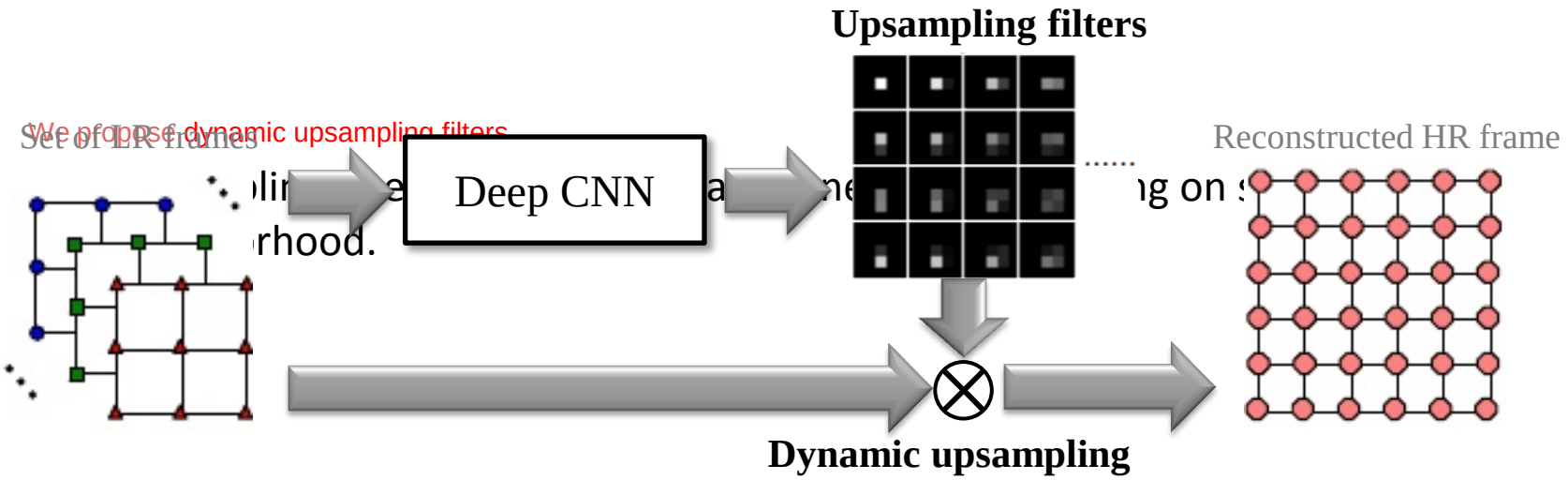


[Ma et al. CVPR 2015]

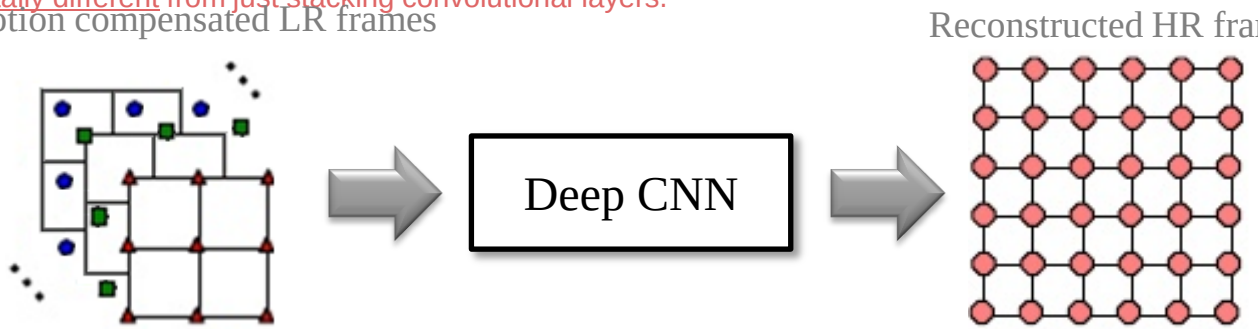
✓ We make our SR network implicitly utilized the motion information.

To Avoid Explicit MC

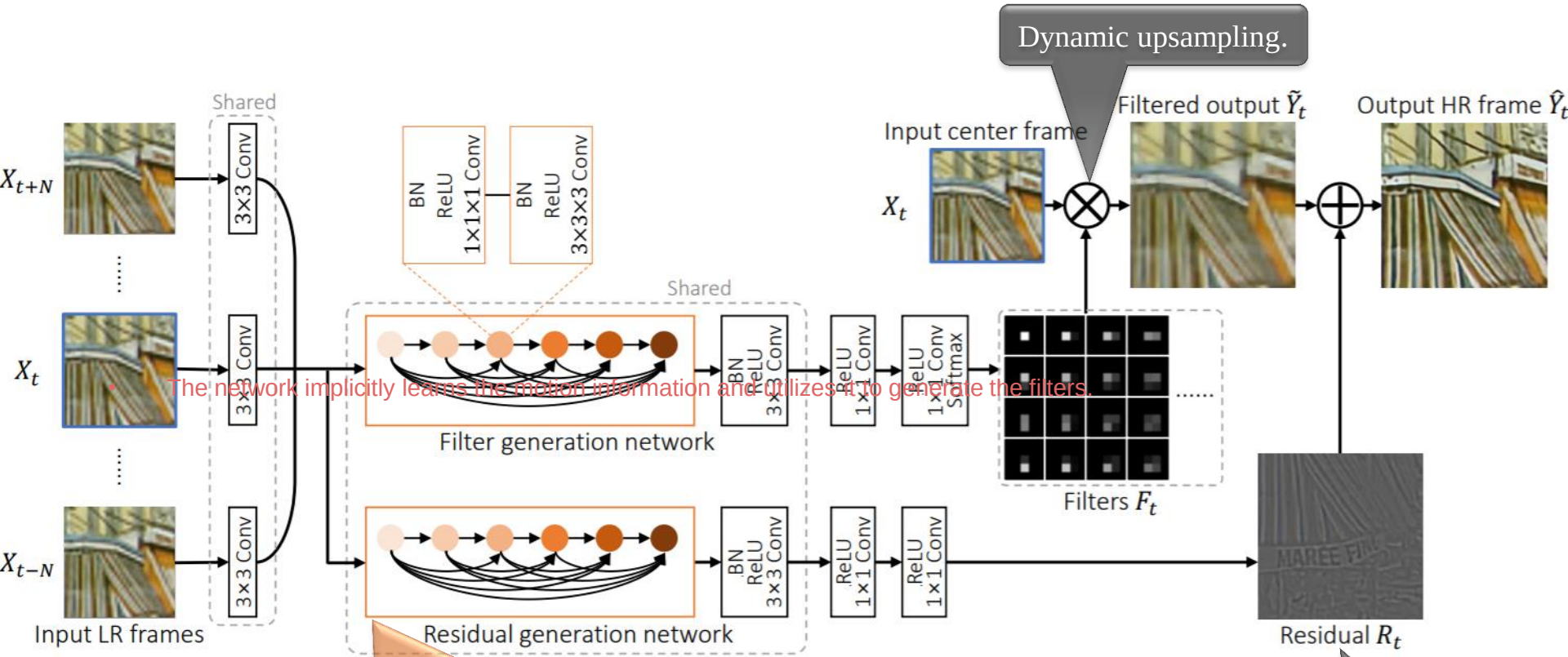
- ✓ We propose dynamic upsampling filters



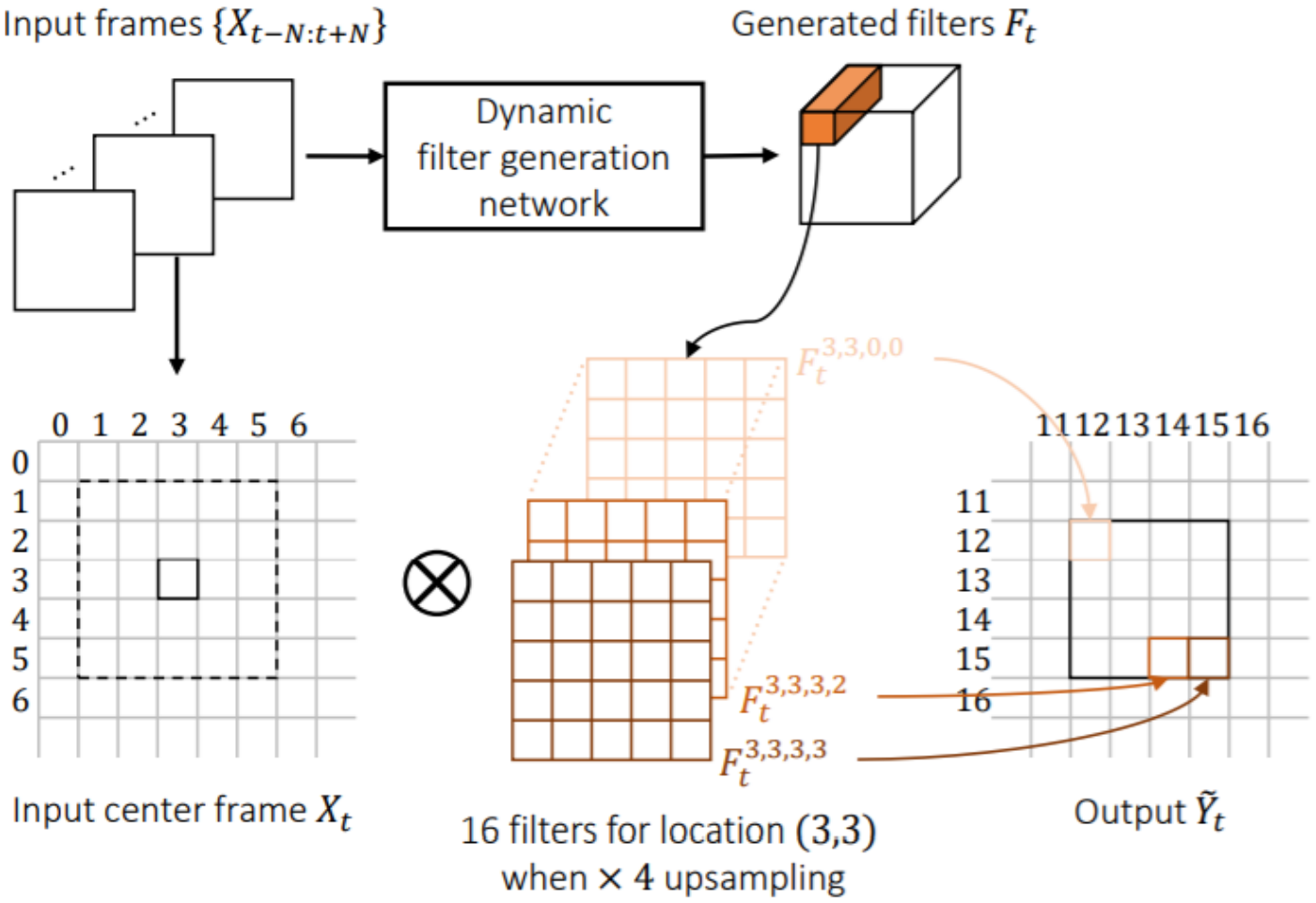
- Fundamentally different from just stacking convolutional layers.



Network Architecture

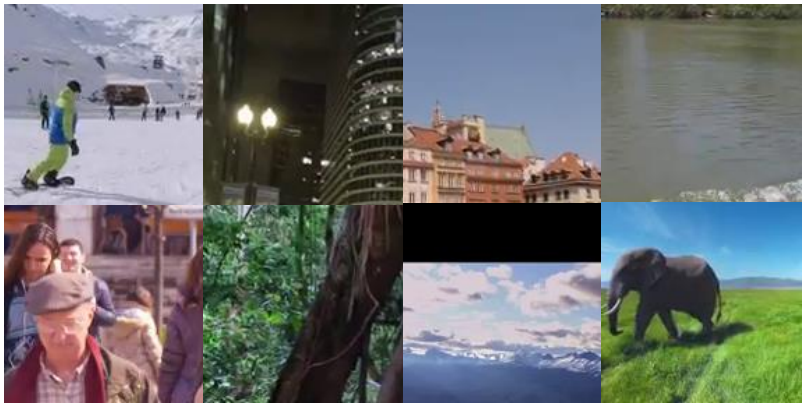


Dynamic Upsampling Filters

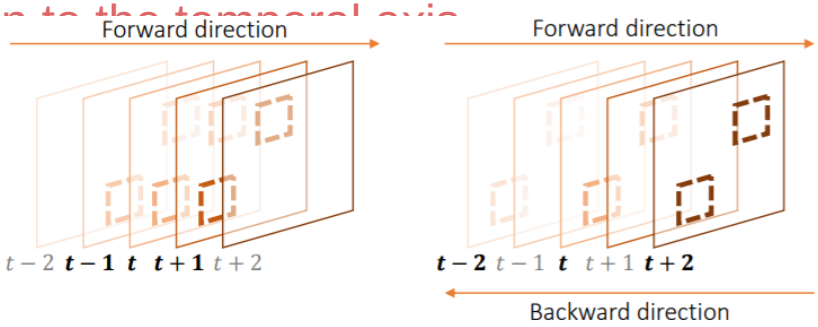


An example of upscaling a pixel at location (3,3) in the center input frame X_t by the upscaling factor $r = 4$. 16 generated filters from $F_t^{3,3,0,0}$ to $F_t^{3,3,3,3}$ are used to create 16 pixels in the region (12,12) to (15,15) of the filtered HR frame $\tilde{Y}_t$.

Datasets

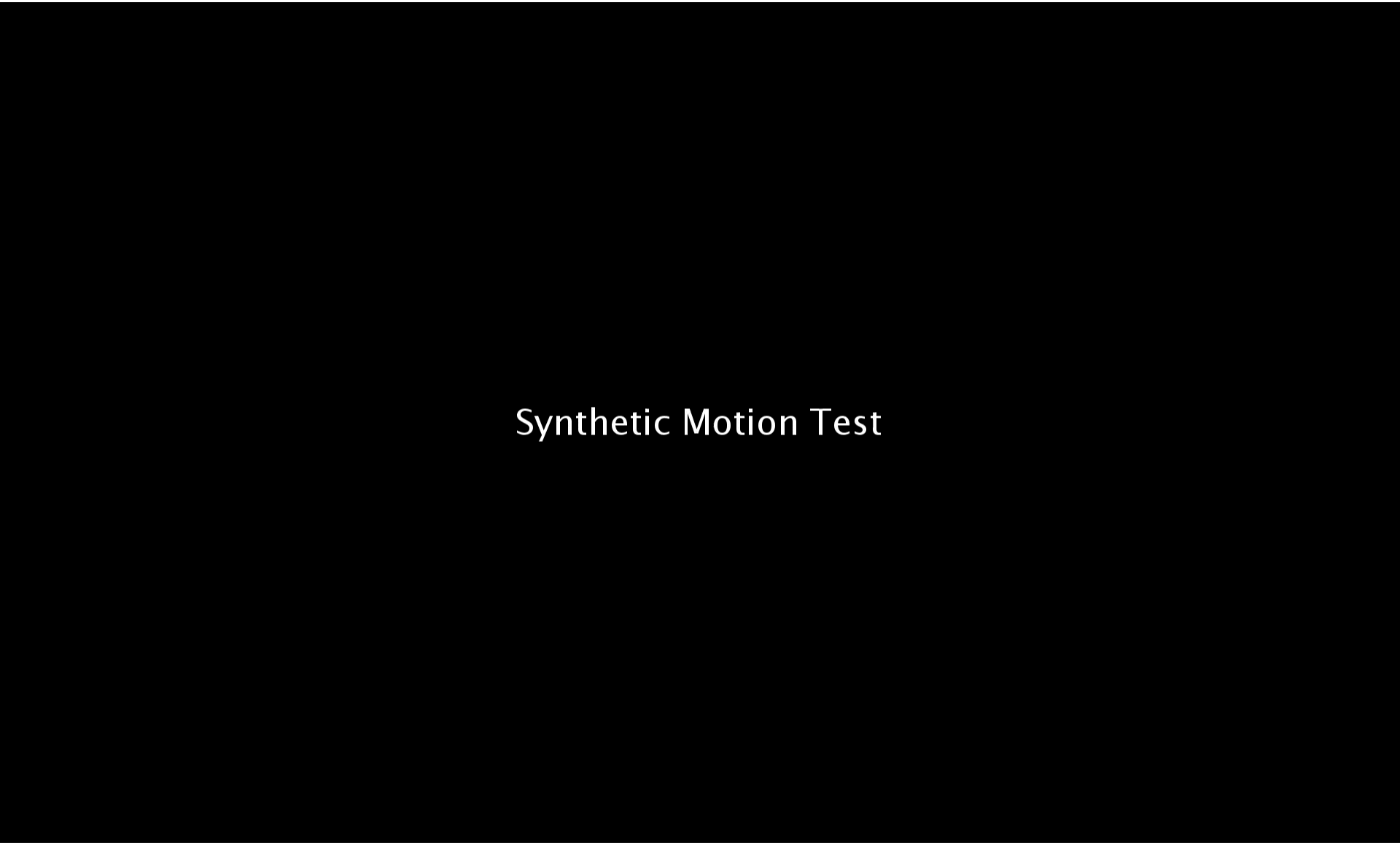


- Training dataset.
  - 351 videos including wildlife, activity, and landscape, which contain various real-world motions and textures.
  - Total 160,000 training data with sufficient amount of motion is extracted from them.
- Data augmentatic[illegible] to the temporal axis
  - Reverse order, s[illegible]



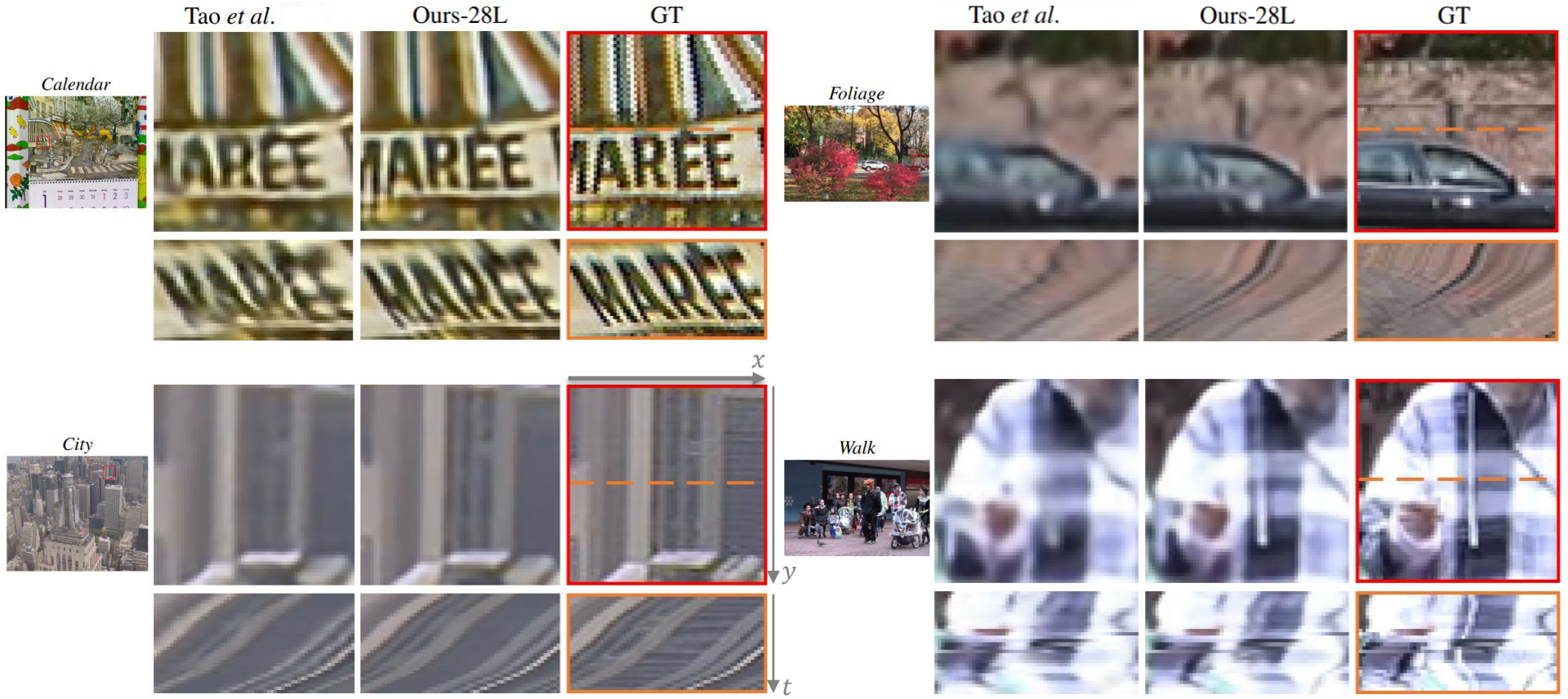
- Test dat[illegible]

Synthetic Motion Test



Comparisons with Other Methods

Upscale	Metric	Bicubic	VSRnet	VESPCN	Tao *et al.*	Liu *et al.*	Ours-10L	Ours-16L	Ours-28L
×2	PSNR	28.43	31.30	-	-	-	**33.73**	-	-
	SSIM	0.8676	0.9278	-	-	-	**0.9554**	-	-
×3	PSNR	25.28	26.79	27.25	-	-	**28.90**	-	-
	SSIM	0.7329	0.8098	0.8447	-	-	**0.8898**	-	-
×4	PSNR	23.79	24.84	25.35	26.01	25.24	**26.81**	**26.99**	**27.34**
	SSIM	0.6332	0.7049	0.7557	0.7744	-	**0.8145**	**0.8215**	**0.8327**



Result Video

Qualitative Comparisons

Summary

- **No explicit motion compensation.**
  - Fundamentally different framework for video SR.
  - The motion information is implicitly utilized to generate dynamic upsampling filters instead.
- **Dynamic upsampling filters.**
  - They are generated locally and dynamically depending on spatiotemporal neighborhood of each pixel.
  - HR frame is directly reconstructed by local filtering to the input center frame.
- **Residual learning.**
  - The filtering alone lacks sharpness as it is still a linear operation.
  - To enhance high frequency details.
- **New data augmentation.**
  - To the temporal axis as well as to the spatial axis.

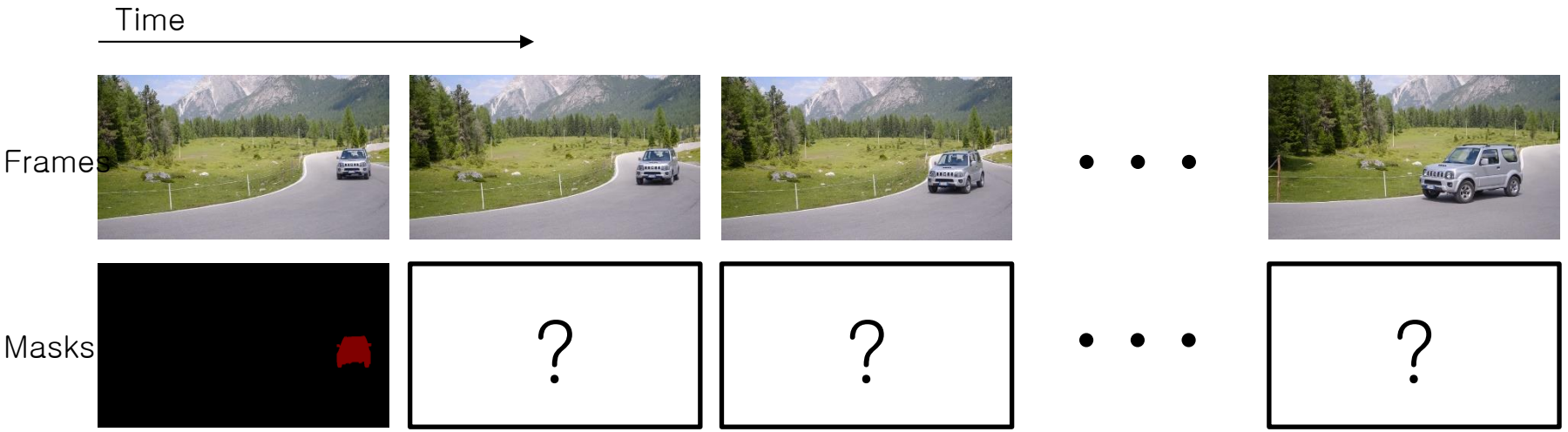
Fast Video Object Segmentation by Reference Guided Mask Propagation

Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, Seon Joo Kim

CVPR 2018

Semi-supervised Video Object Segmentation

Only the first frame annotation is given



Related Works: deep learning based video segmentation

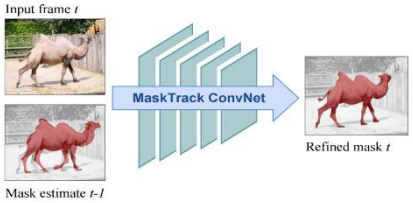
Detection-based methods

- **Strong:** drift and occlusion
- **Weak:** appearance changes

OSVOS
[Caelles et al, CVPR'17]

Propagation-based methods

- **Strong:** appearance changes
- **Weak:** drift and occlusions



MaskTrack
[Perazzi and Khoreva et al, CVPR'17]

Online learning

- Fine-tune network on the target object that appears on the first frame
- **Slow**

Our Approach

Using a **single model,** combine strengths of both lines of approaches:

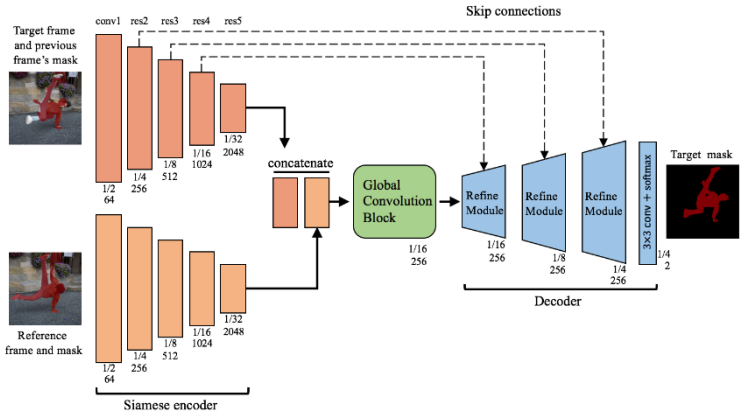
- Appearance of the object (Detection-based)
- Temporal consistency (Propagation-based)

Avoid Online Learning

- Perform object matching **at running time**

Siamese Encoder-Decoder network

- Previous frame's mask (to propagate)
- Reference frame (to detect)



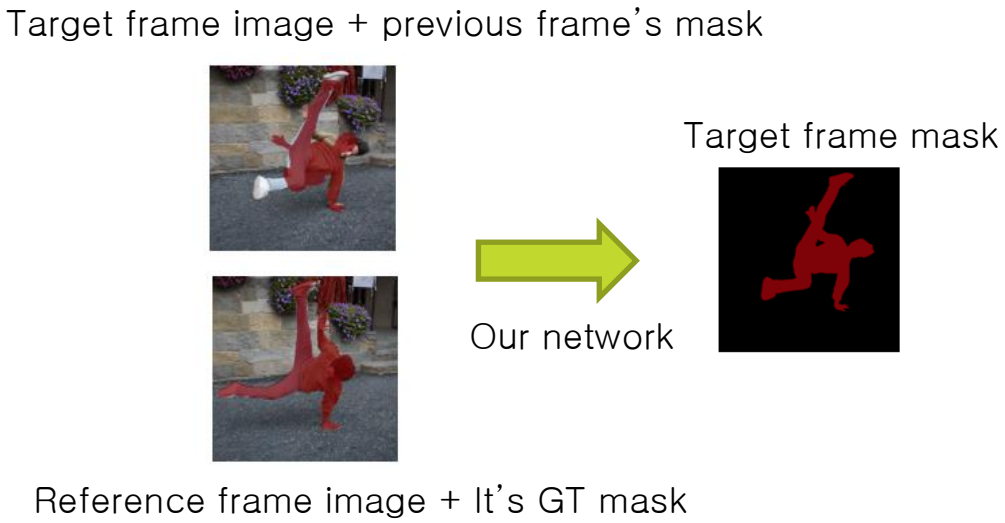
Siamese Encoder-Decoder network

Inputs

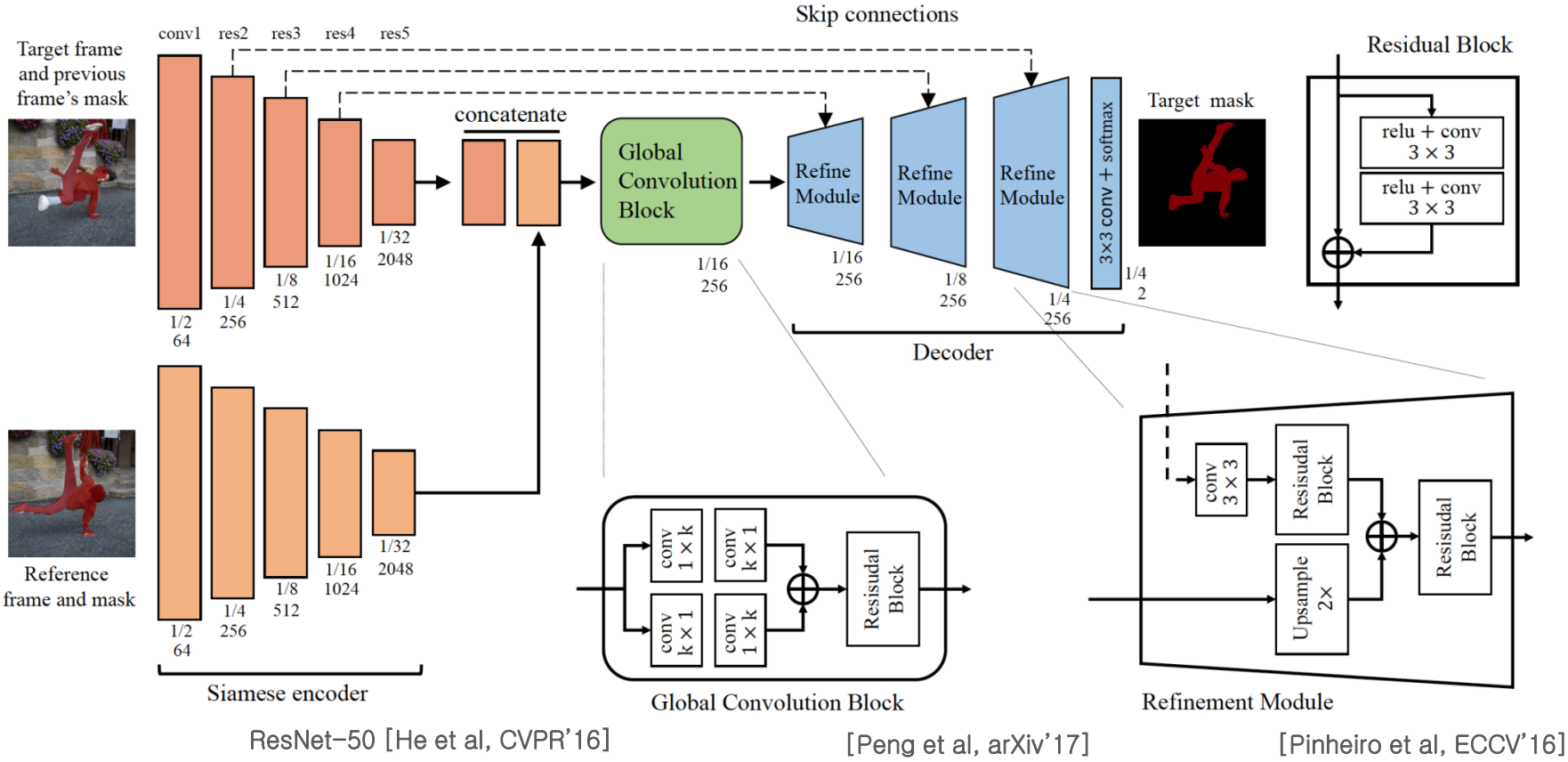
- **(Target stream)**
  - Target frame image
  - previous frame's mask
- **(Reference stream)**
  - Reference frame image
  - it's ground truth mask

Output

- Target frame mask



Siamese Encoder-Decoder network



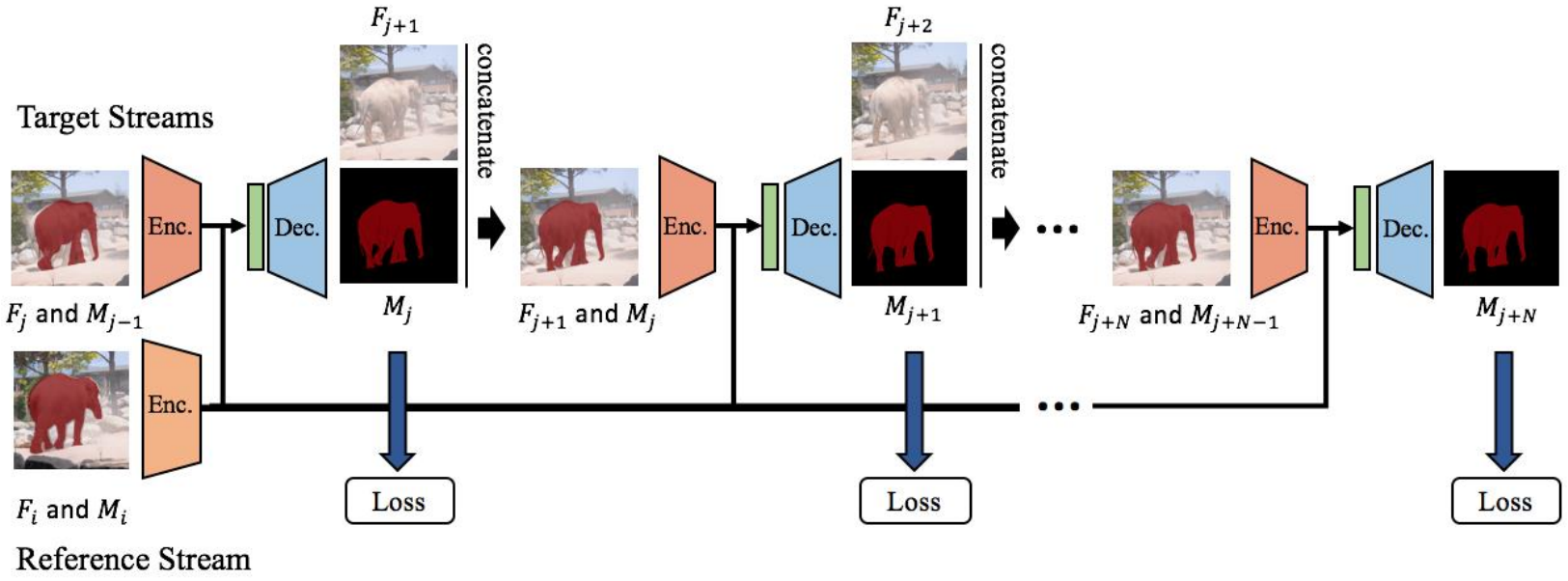
ResNet-50 [He et al, CVPR'16]

[Peng et al, arXiv'17]

[Pinheiro et al, ECCV'16]

Fine-tune on videos: DAVIS-2017

Training with recurrence: mask output is repeatedly used as input for the next estimation.



Back-propagation through time (like RNN)

Two-stage Training

No video segmentation data that provide enough training data

- Recent DAVIS dataset provide 60 short videos for training [Perazzi and J. Pont-Tuset]

First, pre-train using static images

- By **simulating** our network's inputs

Second, fine-tune on videos

Pre-training on static images

Datasets: Pascal VOC, ECSSD, MSRA10K

Generation Method 1:

Random transform

- Realistic but not diverse

Target frame

Reference frame

Pre-training on static images

Datasets: Pascal VOC, ECSSD, MSRA10

Generation Method 1:

Random transform

- Realistic but not diverse

Target frame

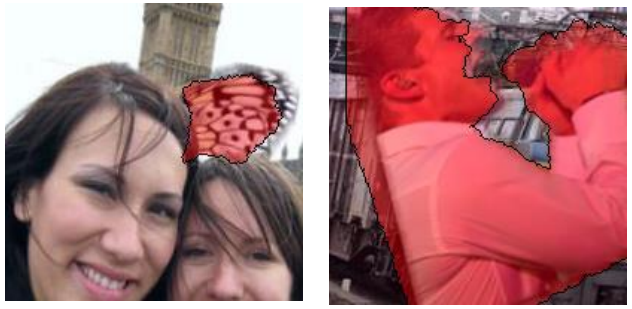
Reference frame

Generation Method 2:

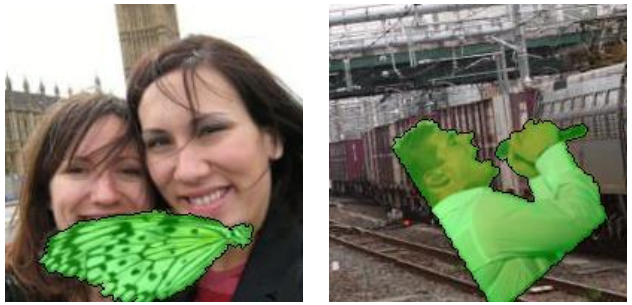
Image composition

- Diverse, but unrealistic

Target frame



Reference frame



Results

No Online learning
No Post-processing

Quantitative results on DAVIS

		VPN	MSK	LCT	PLM	OSVOS	OnAVOS	SegFlow	Ours
Single Object	IoU	70.2	79.7	80.5	70.0	79.8	**86.1**	74.8	**81.5**
	F	65.5	75.4	77.6	62.0	80.6	**84.9**	74.5	**82.0**
Multi Object	IoU	-	-	-	-	52.1	**61.0**	-	**64.8**
	F	-	-	-	-	-	**66.1**	-	**68.6**

Single Object: DAVIS-2016 validation

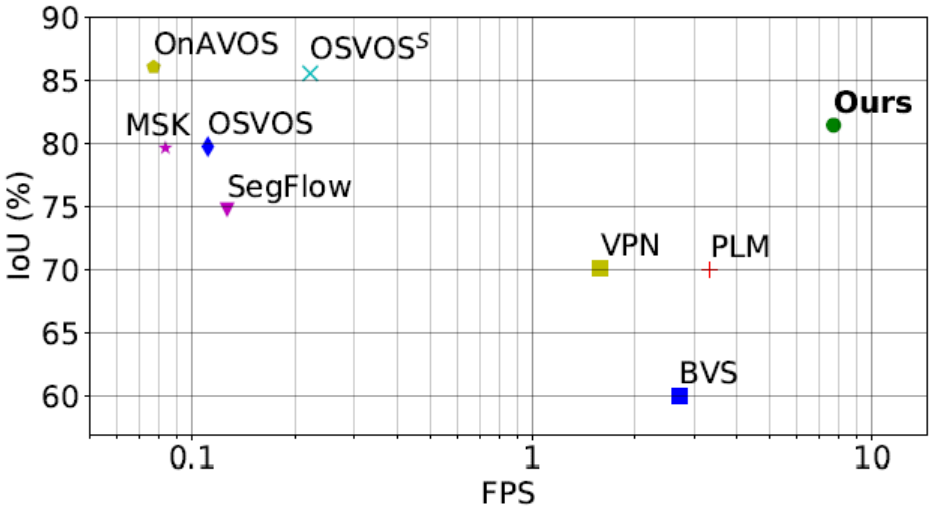
Multiple Object: DAVIS-2017 validation

1st 2nt

Results

No Online learning
No Post-processing

Quantitative results on DAVIS

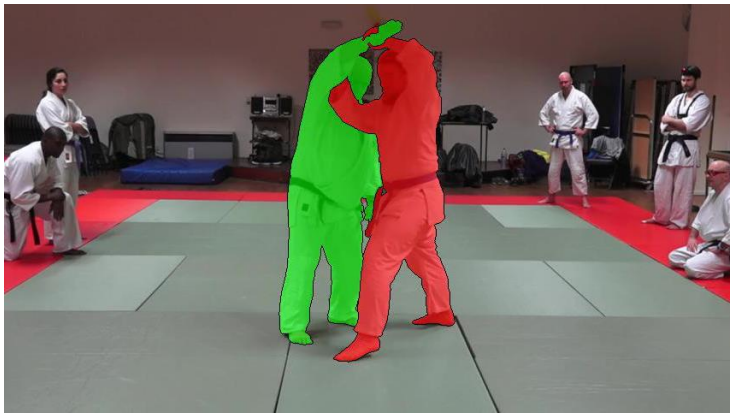


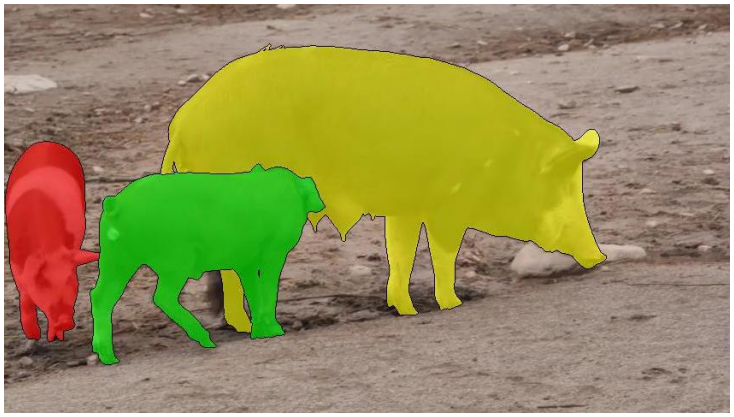
Single Object: DAVIS-2016 validation

Results: Single Object

Results: Single Object

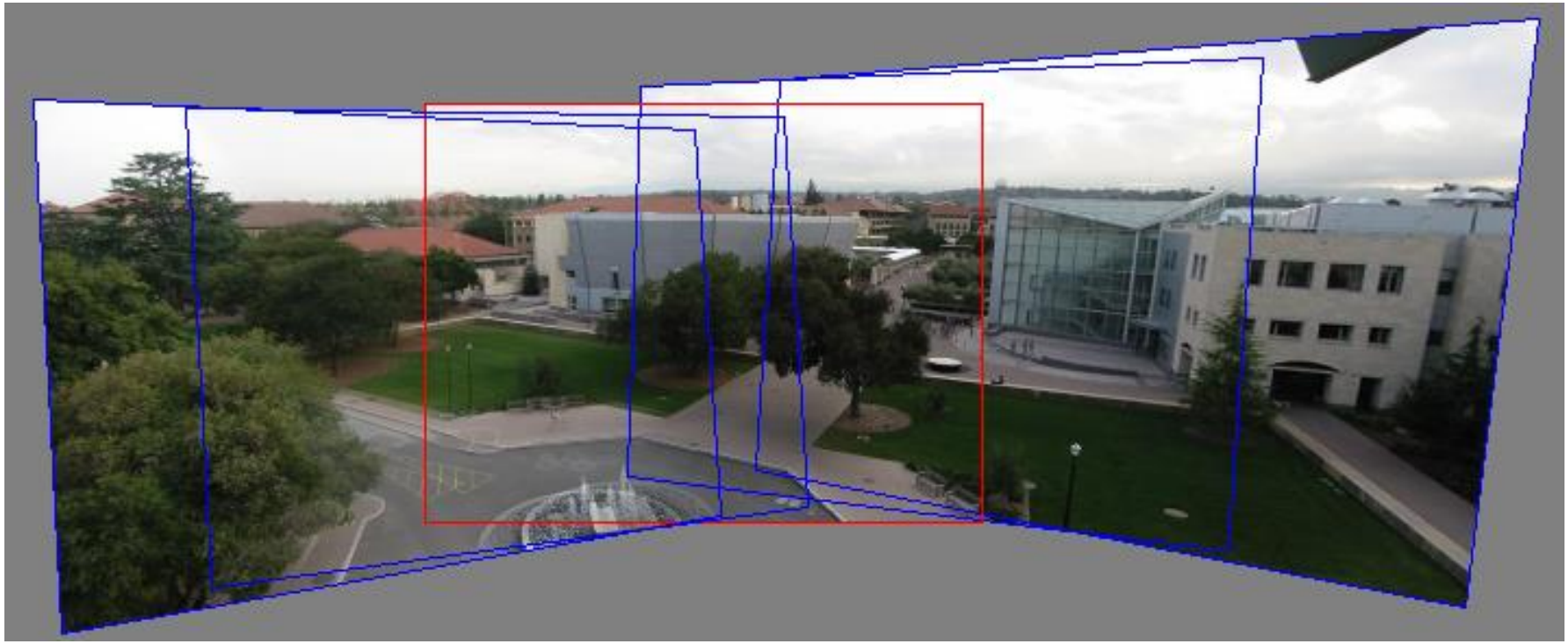
Results: Multiple Object



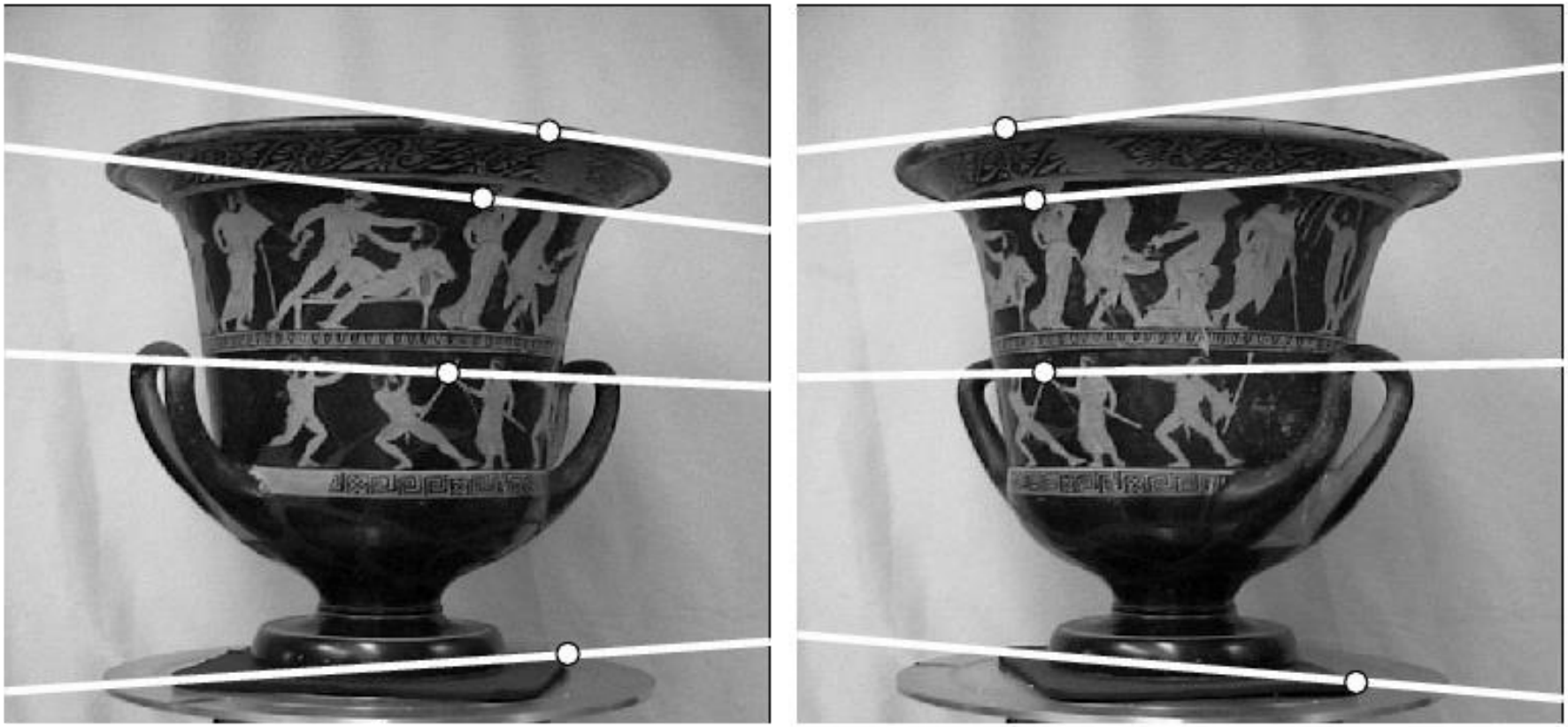


GEOMETRY

Image alignment

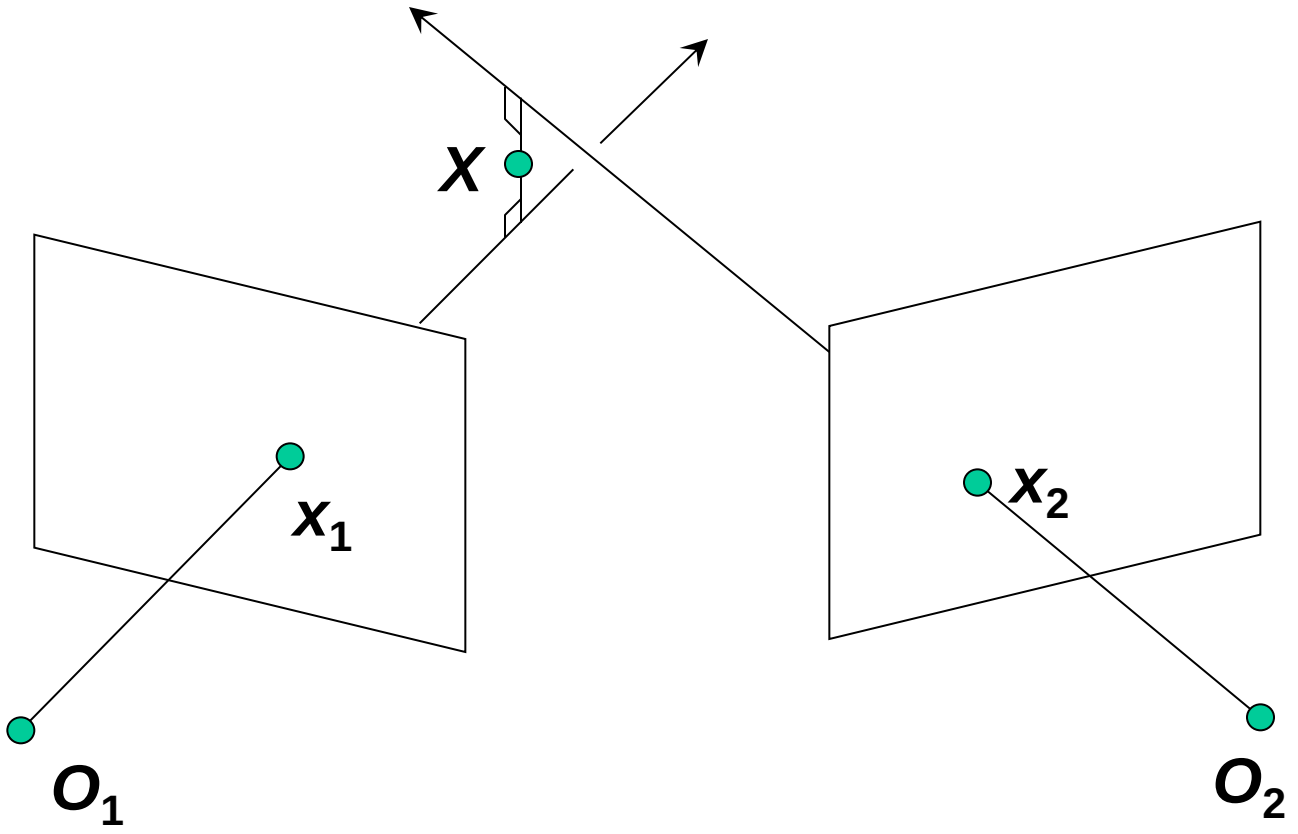


Two-view geometry

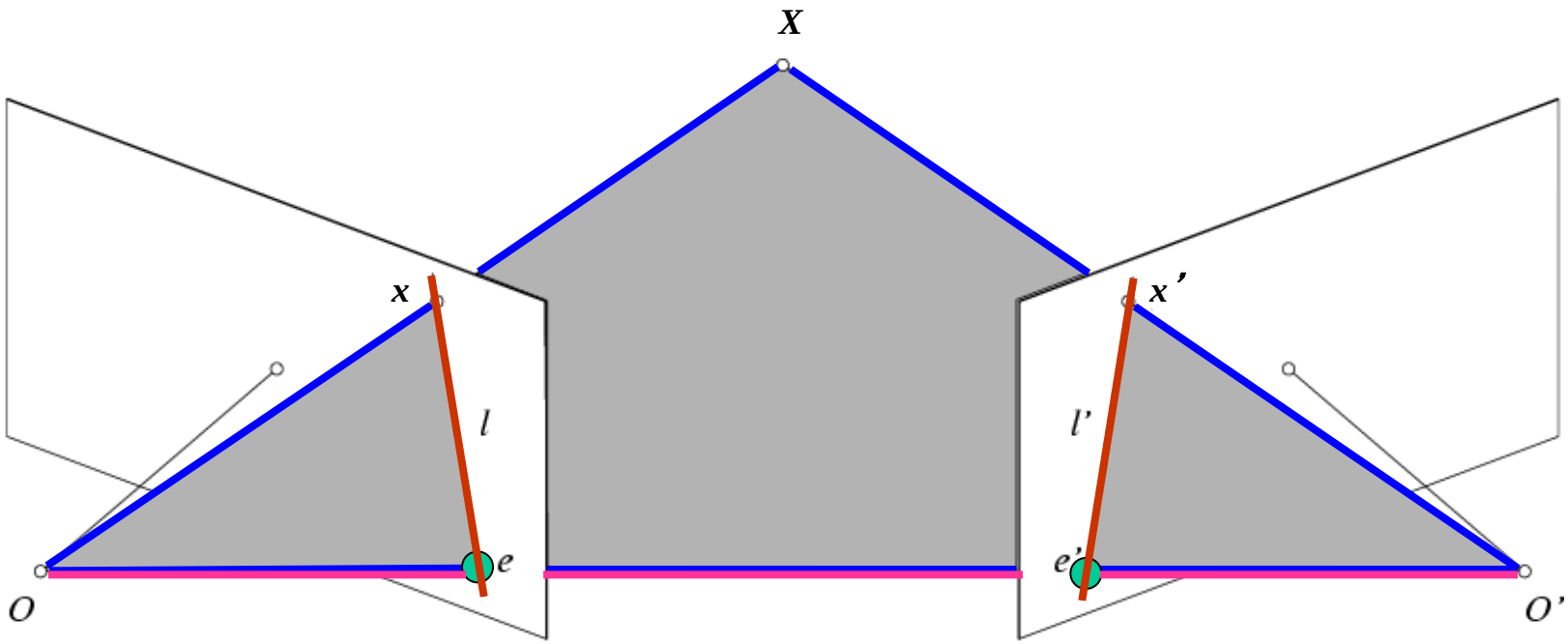


Triangulation: Geometric approach

- Find shortest segment connecting the two viewing rays and let X be the midpoint of that segment



Epipolar geometry



- **Baseline** – line connecting the two camera centers
- **Epipolar Plane** – plane containing baseline
- **Epipoles**

= intersections of baseline with image planes

= projections of the other camera center

Binocular stereo

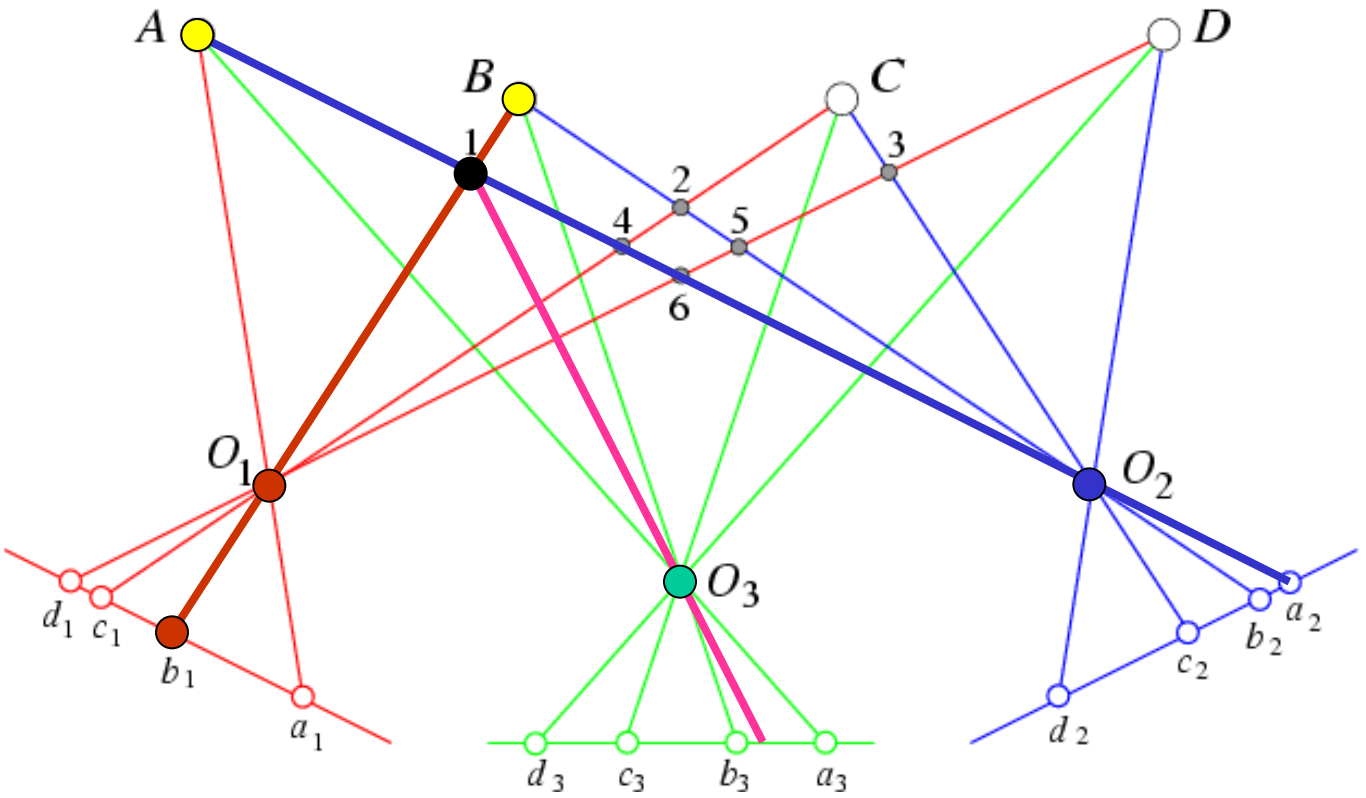
- Given a calibrated binocular stereo pair, fuse it to produce a depth image

image 1

image 2

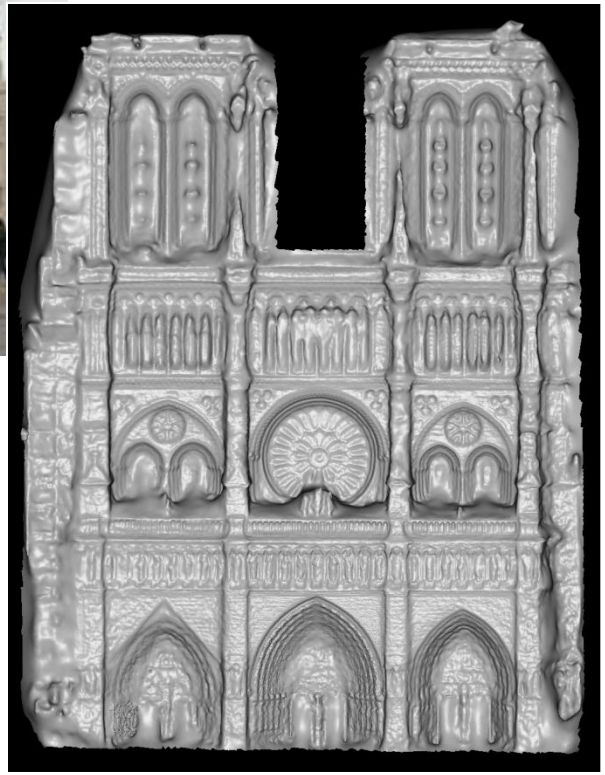
Dense depth map

Beyond two-view stereo



The third view can be used for verification

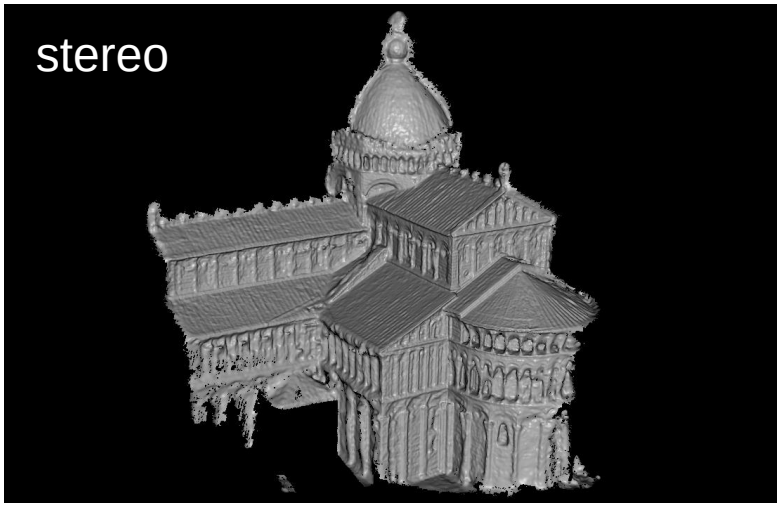
Stereo from community photo collections

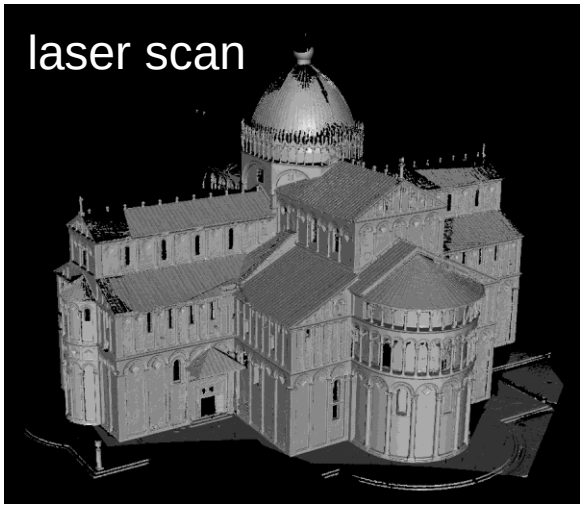


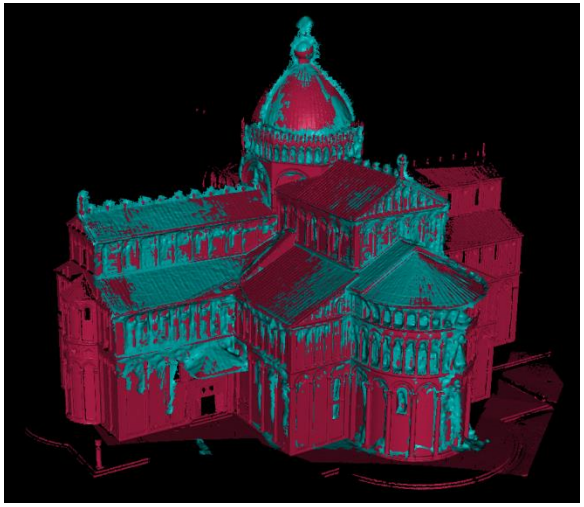
M. Goesele, N. Snavely, B. Curless, H. Hoppe, S. Seitz, Multi-View Stereo for Community Photo Collections, ICCV 2007

http://grail.cs.washington.edu/projects/mvscpc/

Stereo from community photo collections



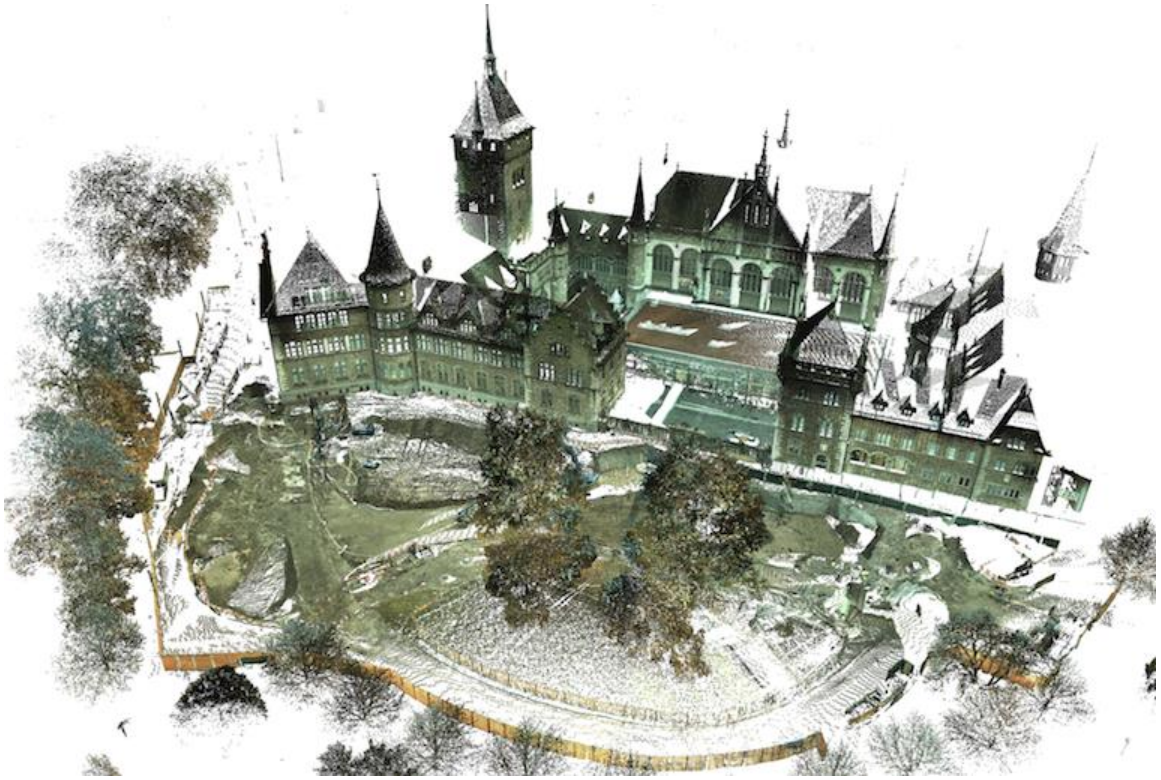




Comparison: 90% of points within 0.128 m of laser scan (building height 51m)

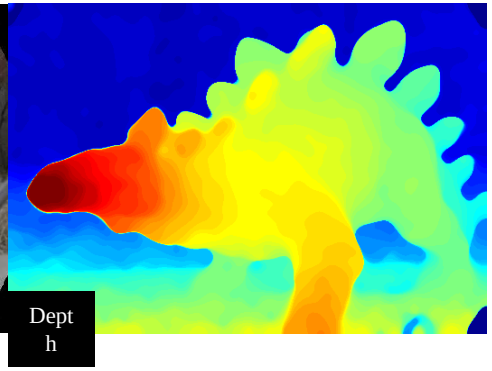
M. Goesele, N. Snavely, B. Curless, H. Hoppe, S. Seitz, Multi-View Stereo for Community Photo Collections, ICCV 2007

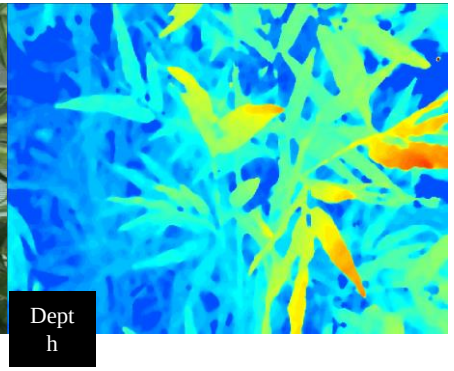
Recent Advances in Geometry in Vision



http://www.youtube.com/watch?v=36PFT6SkYMI

EPINET: A Fully-Convolutional Neural Network using Epipolar Geometry for Depth from Light Field Images





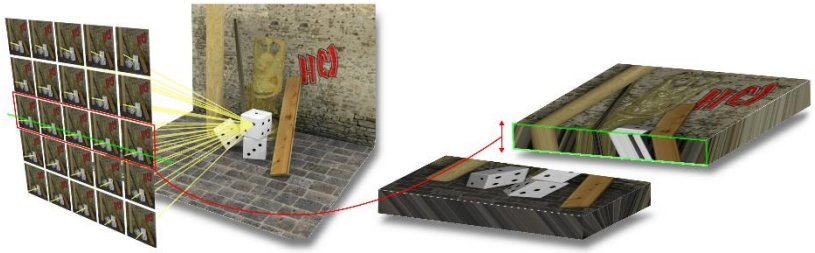
CVPR2018

Changha Shin, Hae Gon Jeon, Young Jin Yoon, In-So Kweon, Seon Joo Kim

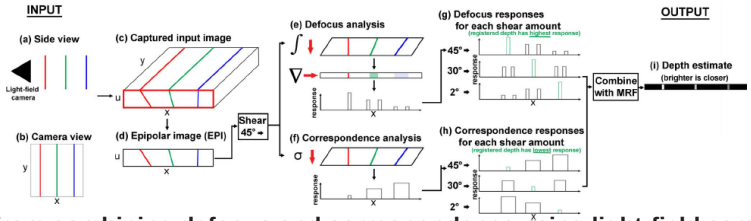
Yonsei University

Which viewpoints are selected to estimate disparity map?

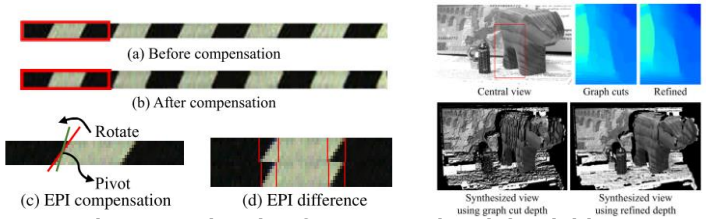
- Total N x N viewpoints



"The Variational Structure of Disparity and Regularization of 4D Light Fields."
B. Goldluecke, S. Wanner *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.



"Depth from combining defocus and correspondence using light-field cameras."
Tao, Michael W., et al**.** *2013 IEEE International Conference on*. IEEE, *Computer Vision*



"Accurate Depth Map Estimation from a Lenslet Light Field Camera."
Hae-Gon Jeon., et al**.** *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Our Network Architecture

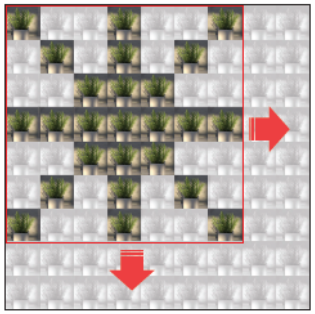
- Design multi-stream networks which can preserve the **Epipolar Property** of light field image.

Table 1. The effect of the number of viewpoints on performance.

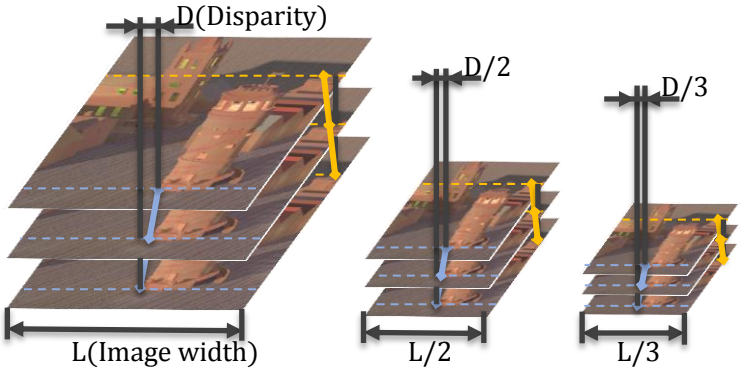
	1-stream	2-streams	4-streams
Input Views			
MSE	2.165	1.729	**1.393**
Bad pixel ratio (<0.07px)	7.61	5.94	**3.87**

Data Augmentation Techniques for LF images

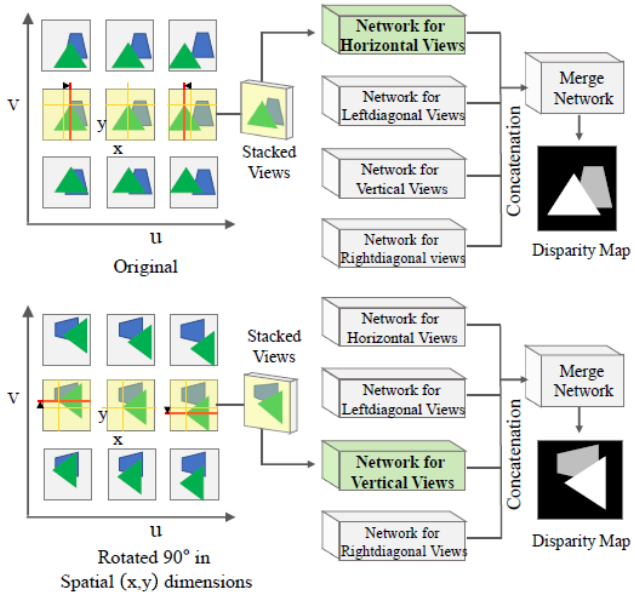
- View-shift augmentation



- Scale augmentation



- Rotation augmentation

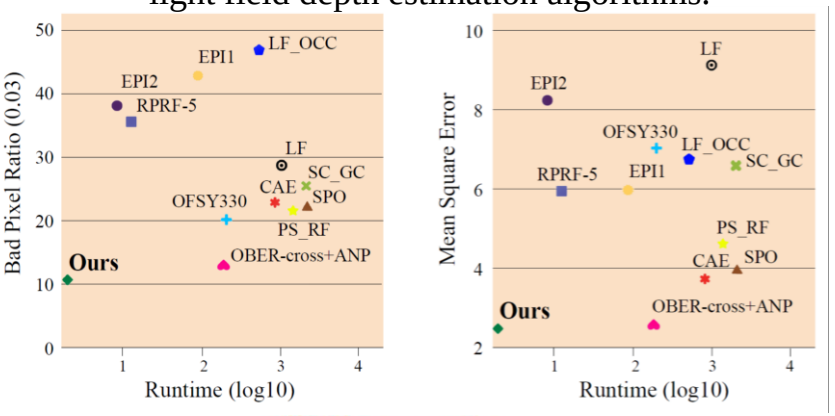


Augular resolution	3 × 3	5 × 5	7 × 7				9 × 9	
Augmentaion type	Full Aug	Full Aug	Color	Color + Viewshift	Color + Rotation	Color + scaling	Full Aug	Full Aug
Mean square error	1.568	1.475	2.799	2.564	1.685	2.33	1.434	1.461
Bad pixel ratio (>0.07px)	8.63	4.96	6.67	6.29	5.54	5.69	3.94	3.91

Results

- 4D Light Field Benchmark
  - 16 synthetic light-field images(9x9 viewpoints)
  - Depth/disparity map for training scenes
  - Site: http://hci-lightfield.iwr.uni-heidelberg.de

Comparison of the accuracy and the runtime of light field depth estimation algorithms.



Bad pixel (Error<0.03)

Algorithm	Meta MEDIAN No preview		Meta AVG No preview	
Epinet-fcn-m	7.731	1	9.537	1
Epinet-fcn	9.501	3	10.745	2
Epinet-fcn9x9	9.058	2	11.212	3
OBER-cross+ANP	11.018	4	13.100	4
SPO-MO	15.243	5	14.258	5
OBER-cross	15.465	6	18.731	6
OFSY_330/DNR	20.373	8	20.225	7
PS_RF	19.719	7	21.630	8
SPO	25.215	15	22.300	9
GLFCV	23.479	11	22.450	10
ZCTV	25.721	16	22.917	11
CAE	23.386	10	22.949	12
RM3DE	23.561	12	23.259	13
OBER	23.713	13	23.565	14
EPN+OS+GC	21.731	9	23.828	15

Mean Square Error (multiplied with 100)

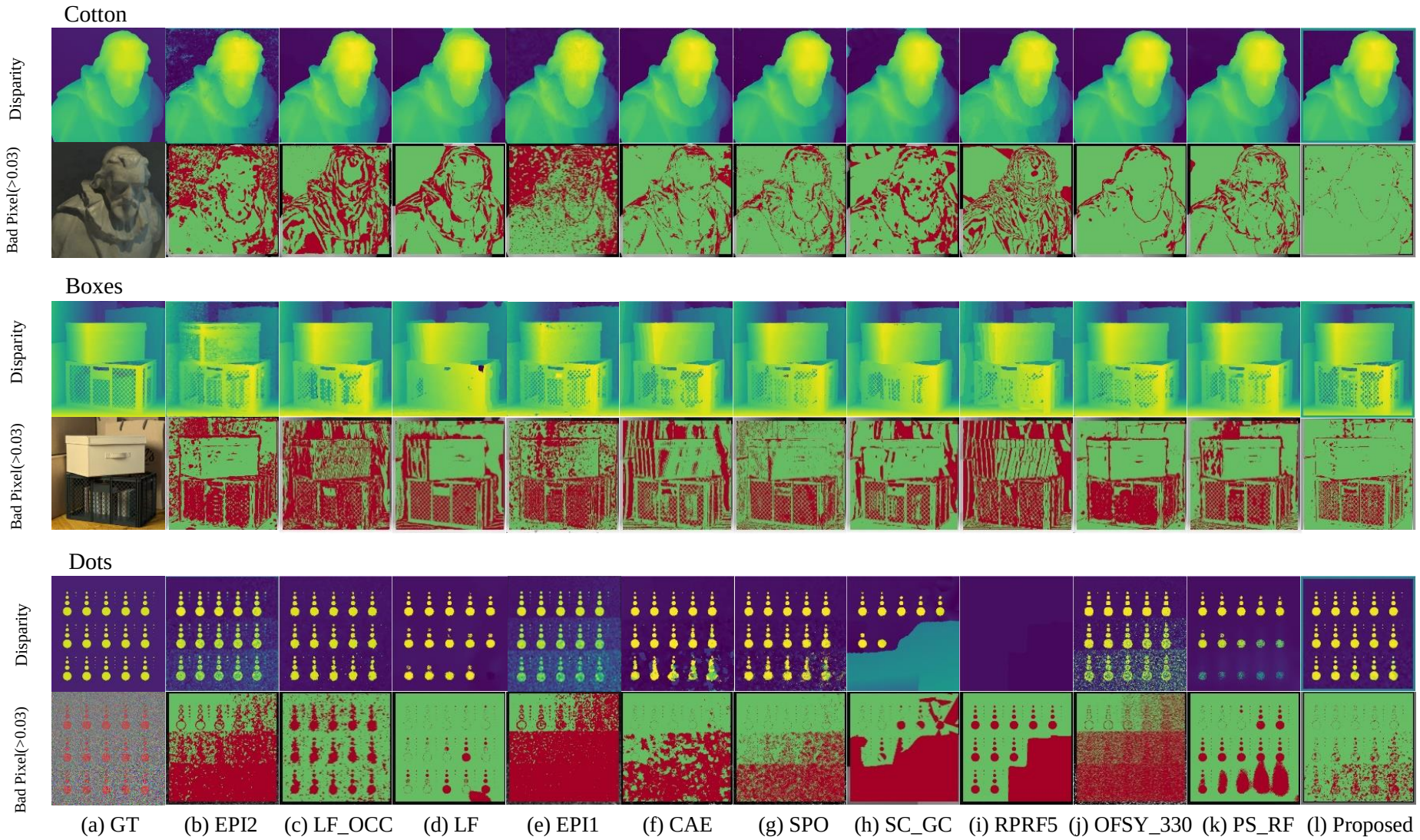
Algorithm	Meta MEDIAN No preview		Meta AVG No preview	
Epinet-fcn-m	1.203	1	2.418	1
Epinet-fcn	1.208	2	2.476	2
Epinet-fcn9x9	1.280	3	2.521	3
OBER-cross+ANP	1.464	5	2.584	4
SPO-MO	1.805	7	3.518	5
CAE	2.667	11	3.730	6
FBS*	1.701	6	3.805	7
RM3DE	1.455	4	3.922	8
SPO	3.309	15	3.968	9
OBER-cross	2.547	10	4.010	10
OBER	2.381	9	4.616	11
PS_RF	2.169	8	4.617	12
RPRF	3.760	17	5.683	13
RPRF-5view	3.295	14	5.948	14
*EPI1	3.932	18	5.975	15

Runtime (seconds)

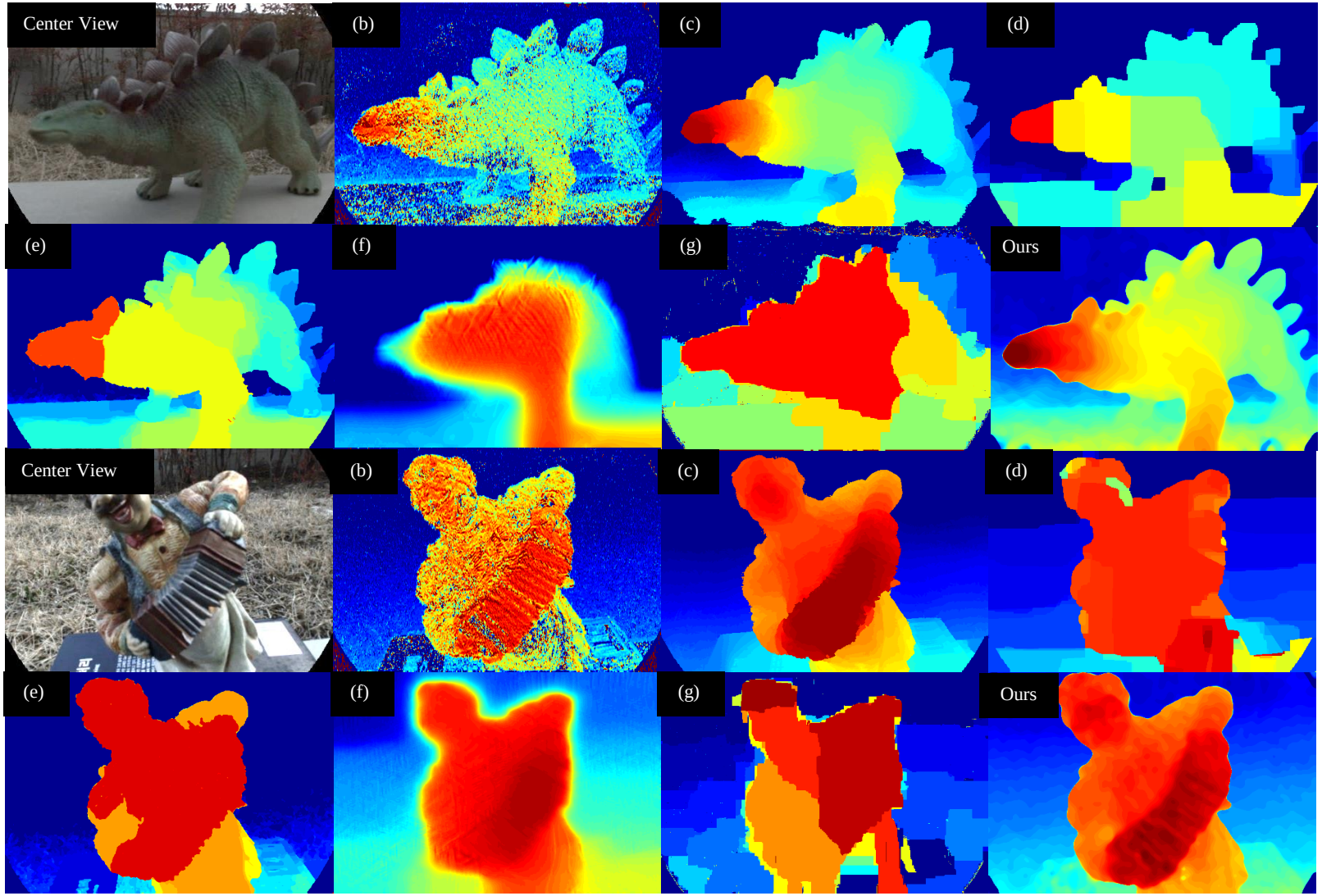
Algorithm	Meta MEDIAN No preview		Meta AVG No preview	
MVCMv0	0.110	1	0.111	1
Epinet-fcn	1.972	2	1.976	2
Epinet-fcn9x9	2.032	3	2.041	3
GLFCV	4.049	4	4.050	4
BSL_I	4.640	5	5.456	5
BSL	5.149	6	5.962	6
*EPI2	8.789	7	8.406	7
Epinet-fcn-m	10.647	8	10.659	8
RPRF-5view	12.966	9	12.498	9
RPRF	35.456	10	34.529	10
ZCTV	37.400	11	37.450	11
OBER	40.420	12	40.141	12
*MV	45.638	14	40.879	13
RM3DE	45.149	13	47.434	14

Algorithm	Meta MEDIAN No preview		Meta AVG No preview	
*EPI1	85.045	15	88.194	15
OBER-cross	96.200	16	96.417	16
OBER-cross+ANP	174.595	17	182.994	17
OFSY_330/DNR	198.299	18	200.282	18
EPN+OS+GC	260.602	19	274.710	19
LF_OCC	484.883	20	519.868	20
FBS*	680.850	21	655.905	21
CAE	822.272	22	832.081	22
*LF	994.311	23	1009.756	23
OMG_occ	1049.401	24	1163.938	24
PS_RF	1141.815	25	1412.623	25
SC_GC	2052.190	26	2056.344	26
SPO	2111.500	27	2115.417	27
SPO-MO	4297.000	28	4304.333	28

Synthetic Results

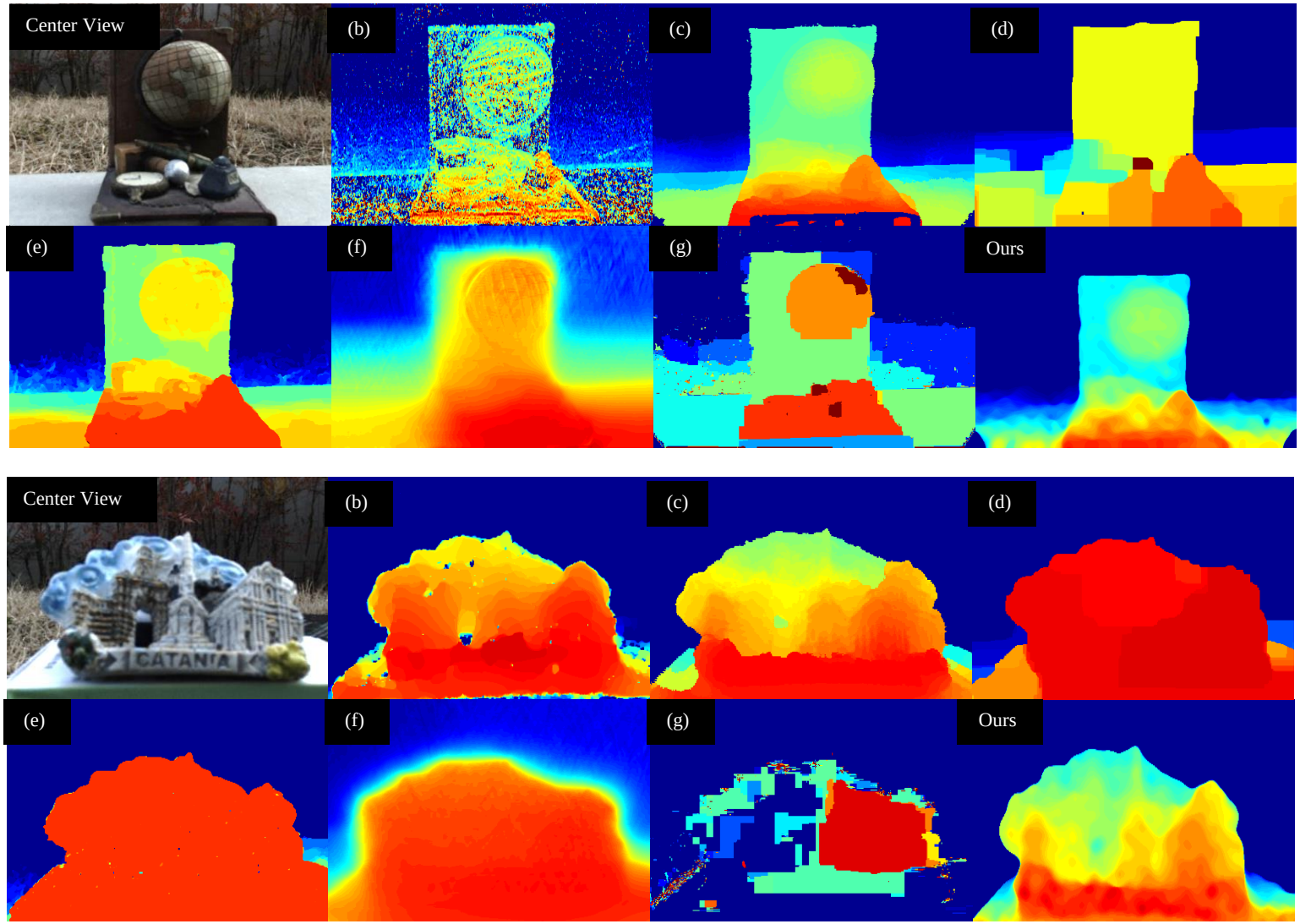


Real-world Results



B: Globally consistent depth labeling of 4D lightfields. S. Wanner and B. Goldluecke
C: Accurate depth map estimation from a lenslet light field camera.H.-G. Jeon,
D: Robust light field depth estimation for noisy scene with occlusion. W. Williem
E: Occlusionaware depth estimation using light-field cameras. T.-C. Wang,.
F: Shape estimation from shading, defocus, and correspondence using light-field angular coherence. Tao
G: Line assisted light field triangulation and stereo matching.Z. Yu,

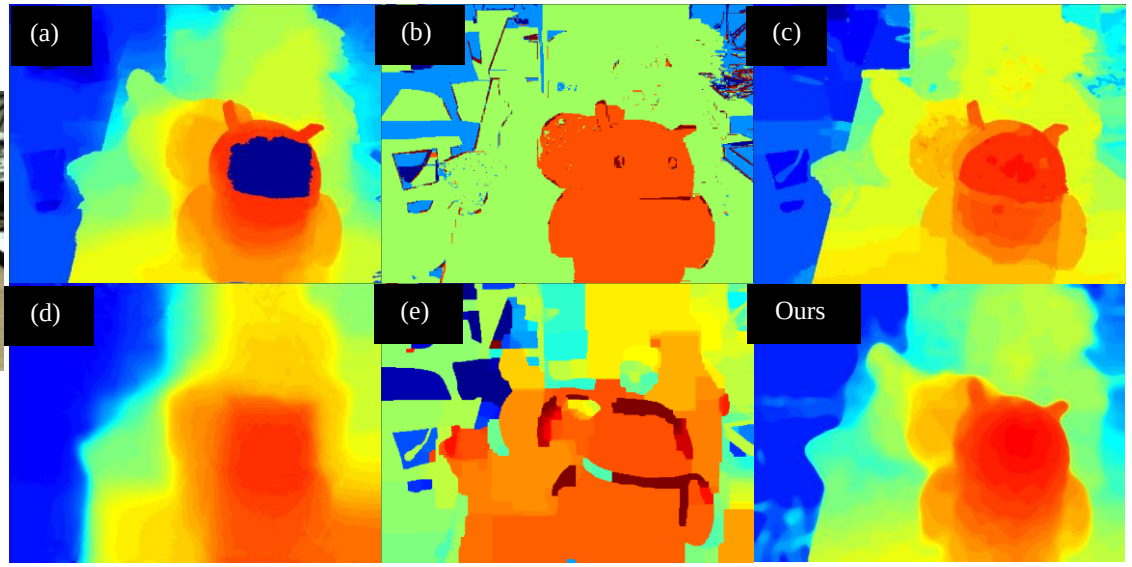
Real-world Results



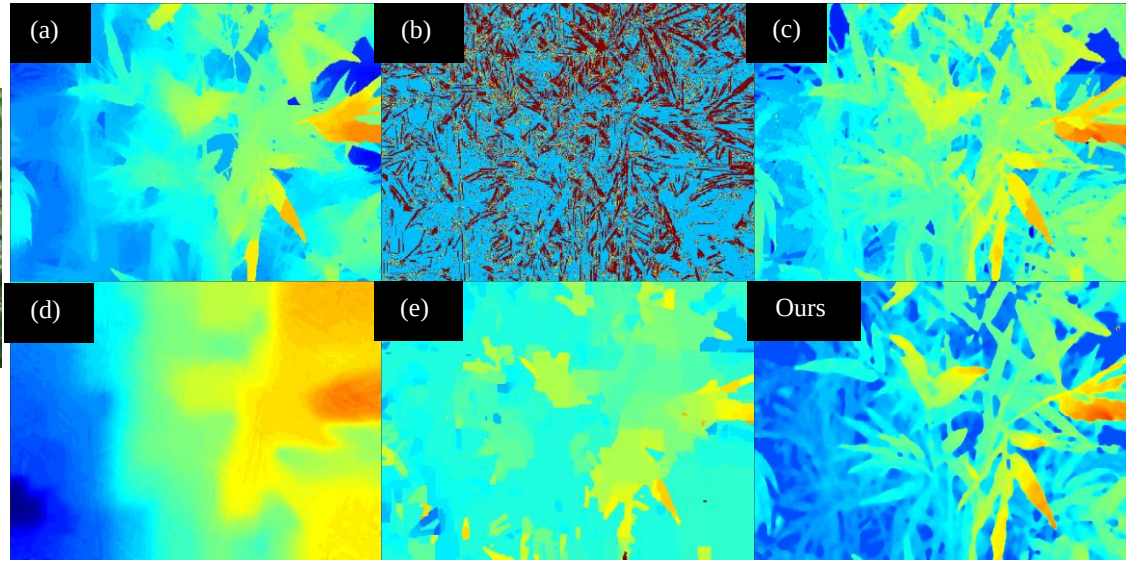
B: Globally consistent depth labeling of 4D lightfields. S. Wanner and B. Goldluecke C: Accurate depth map estimation from a lenslet light field camera.H.-G. Jeon,
D: Robust light field depth estimation for noisy scene with occlusion. W. Williem E: Occlusionaware depth estimation using light-field cameras. T.-C. Wang,.
F: Shape estimation from shading, defocus, and correspondence using light-field angular coherence. Tao G: Line assisted light field triangulation and stereo matching.Z. Yu,

Real-world Results

Center View

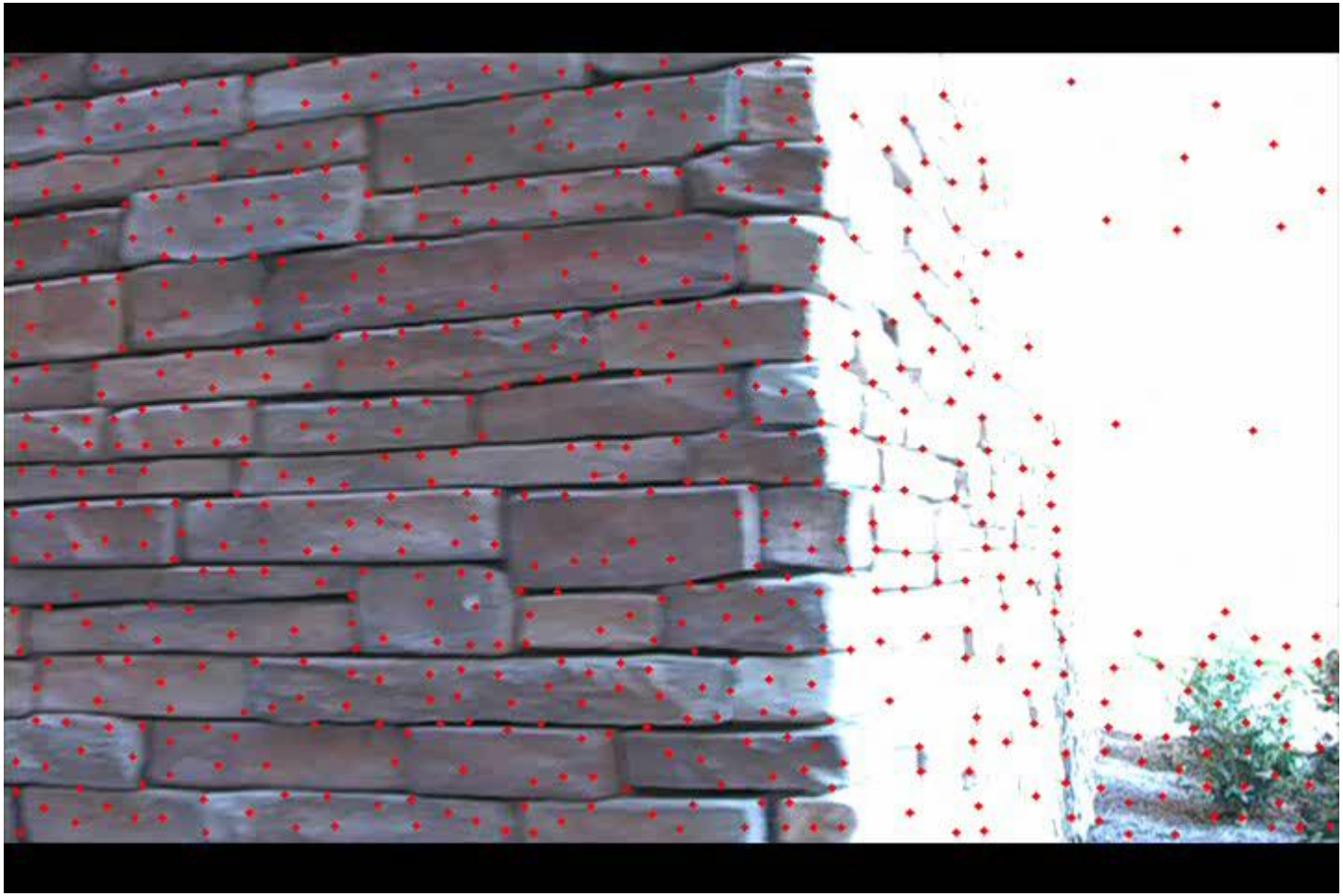


Center View



A: Accurate depth map estimation from a lenslet light field camera.H.-G. Jeon, B: Robust light field depth estimation for noisy scene with occlusion. W. Williem
C: Occlusionaware depth estimation using light-field cameras. T.-C. Wang,. D: Shape estimation from shading, defocus, and correspondence using light-field angular coherence. Tao
E: Line assisted light field triangulation and stereo matching.Z. Yu,

Optical Flow



Optical Flow

- Applications
  - Motion based segmentation
  - Structure from Motion(3D shape and Motion)
  - Alignment (Global motion compensation)
    - Camcorder video stabilization
    - UAV Video Analysis
  - Video Compression

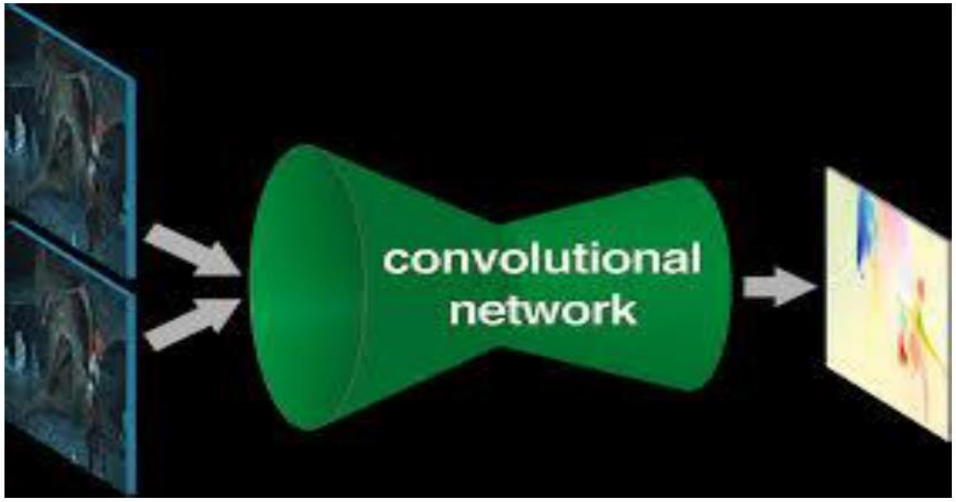
FlowNet: Learning Optical Flow with Convolutional Networks

FischerPhilipp*, *Alexey Dosovitskiy, *Eddy Ilg, **Philip Häusser, **Caner Hazırbaş, **Vladimir Golkov, **Patrick van der Smagt, **Daniel Cremers, and *Thomas Brox

*University of Freiburg, **Technical University of Munich

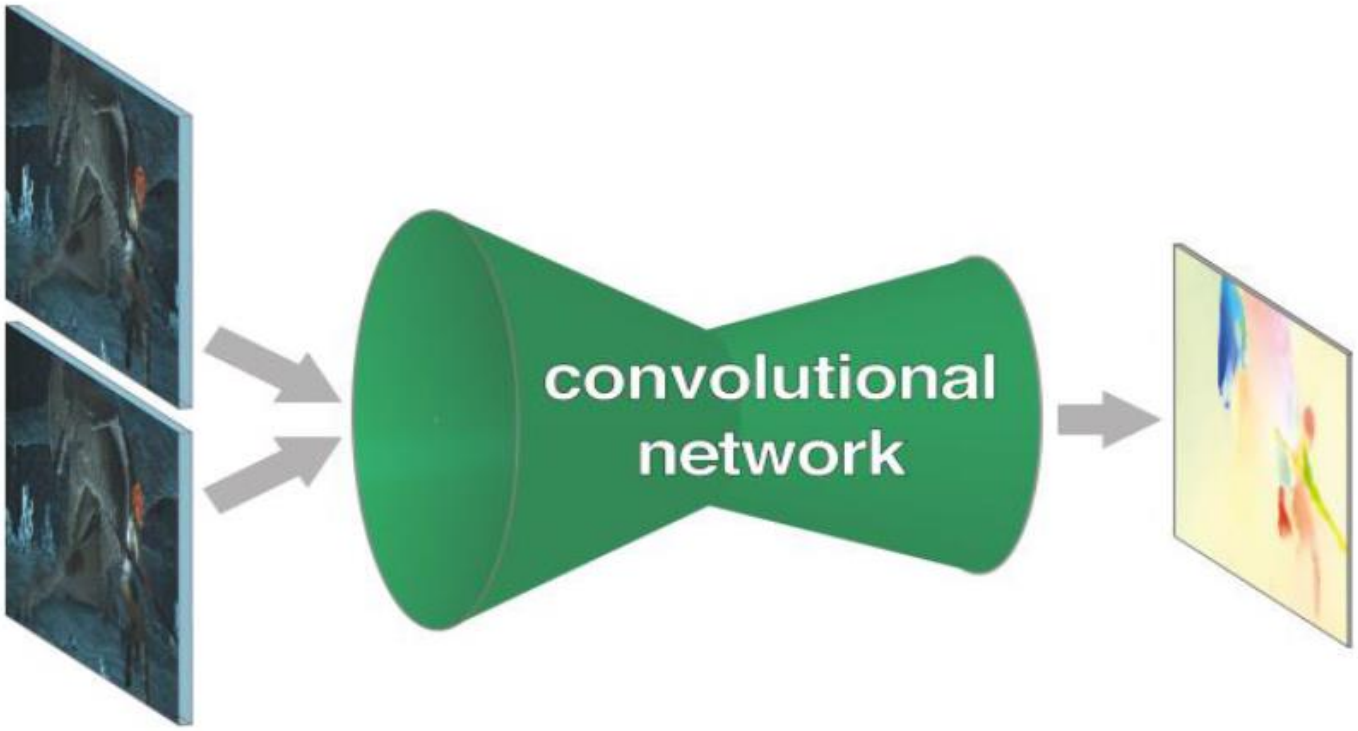
Problem Statement

- Given the pair of images, predict the optical flow field using CNNs.

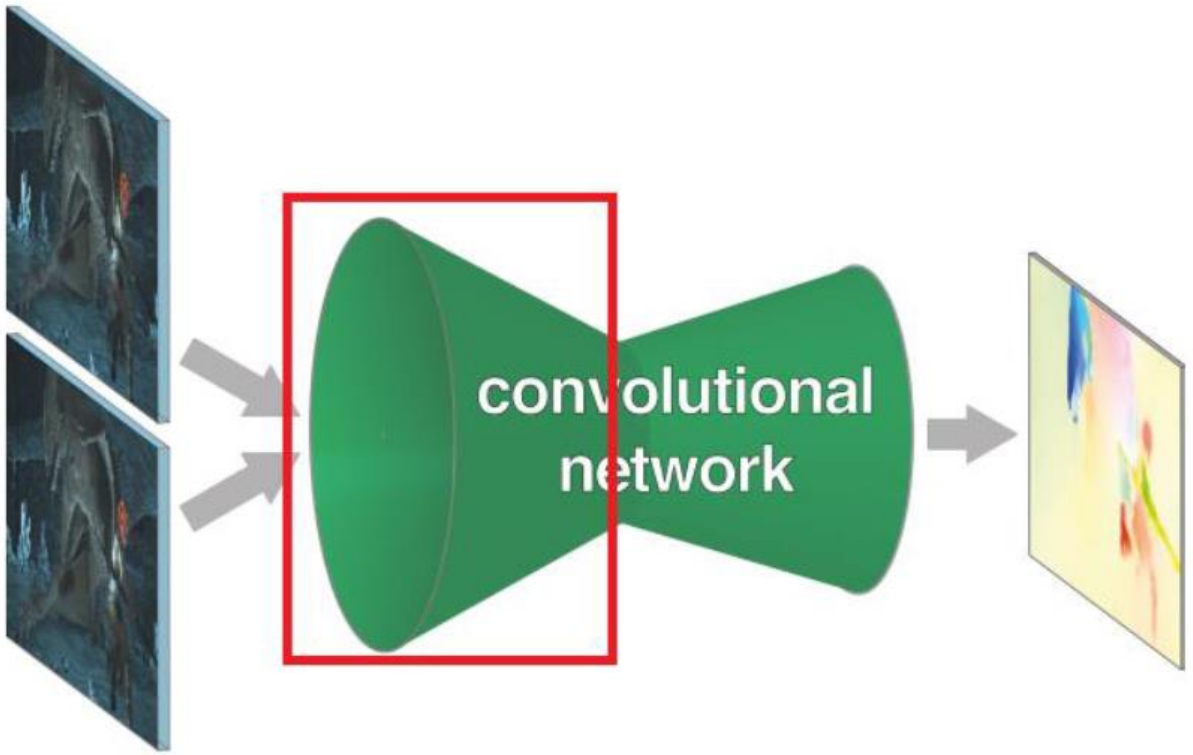


Details of the Proposed Approach

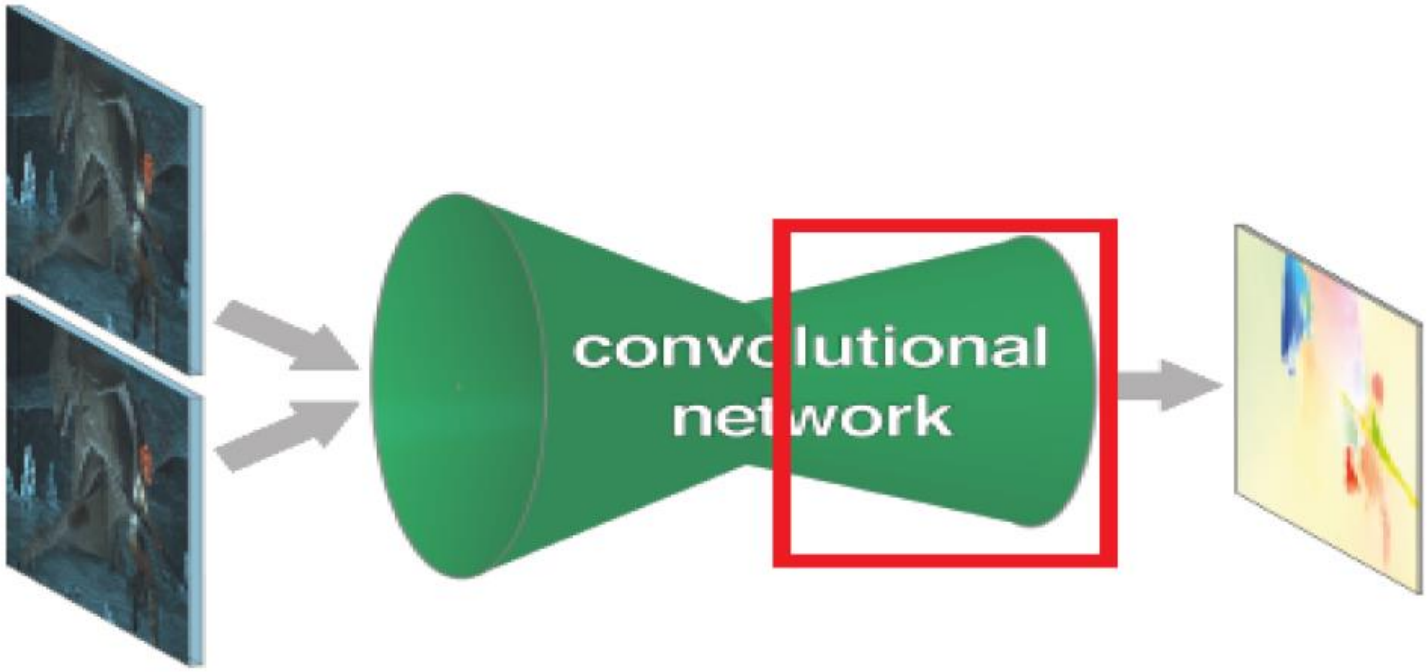
- The network is trained end-to-end.
- Contracting part and expanding part



- Contracting part extracts a rich feature representation

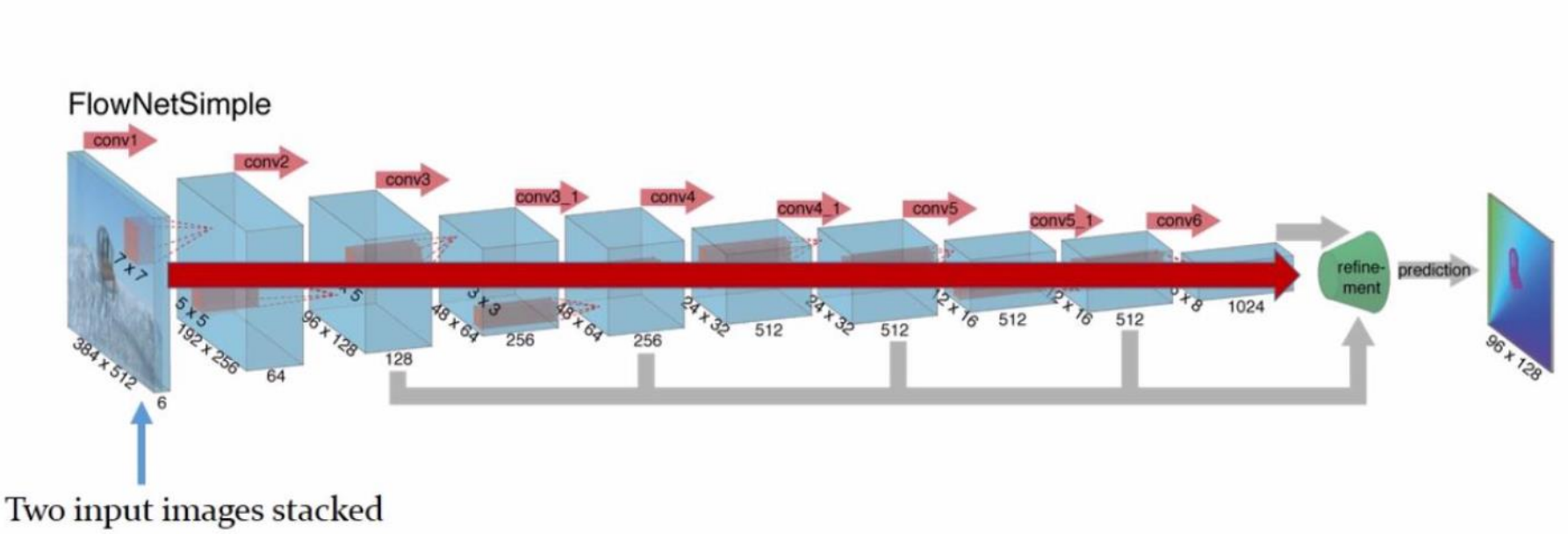


- Expanding Part
  - Refinement
  - Produces the high resolution flow



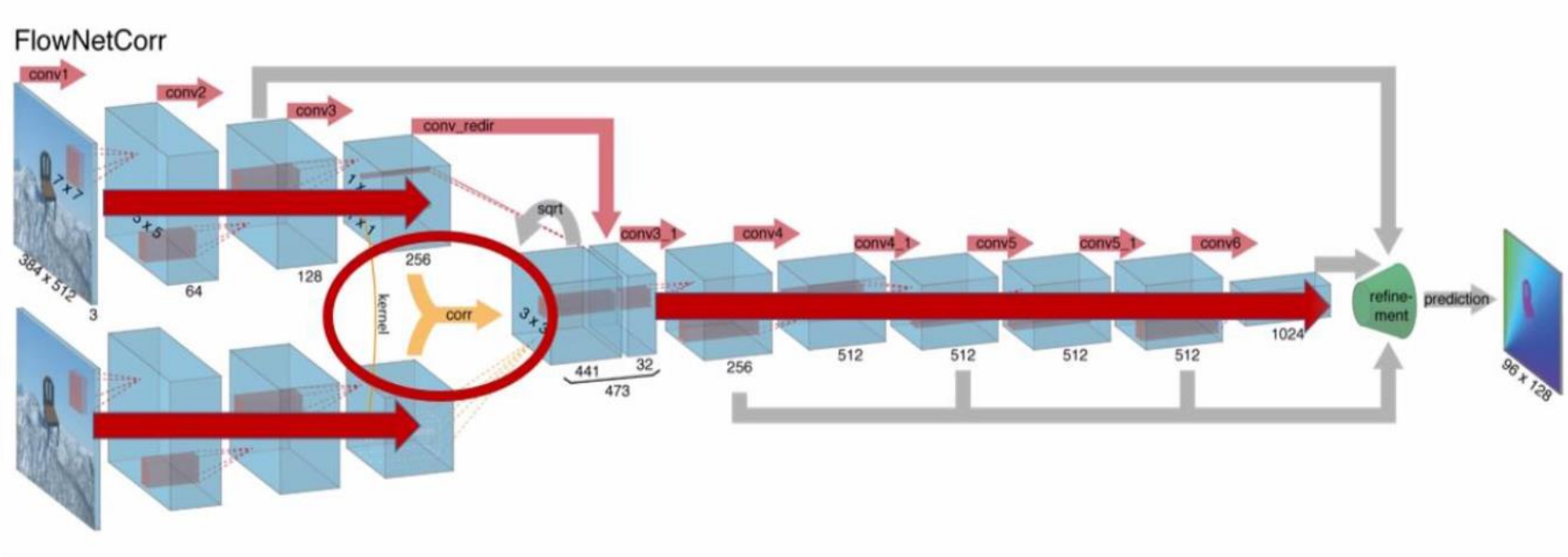
FlowNetSimple

- Process two stacked input images jointly.



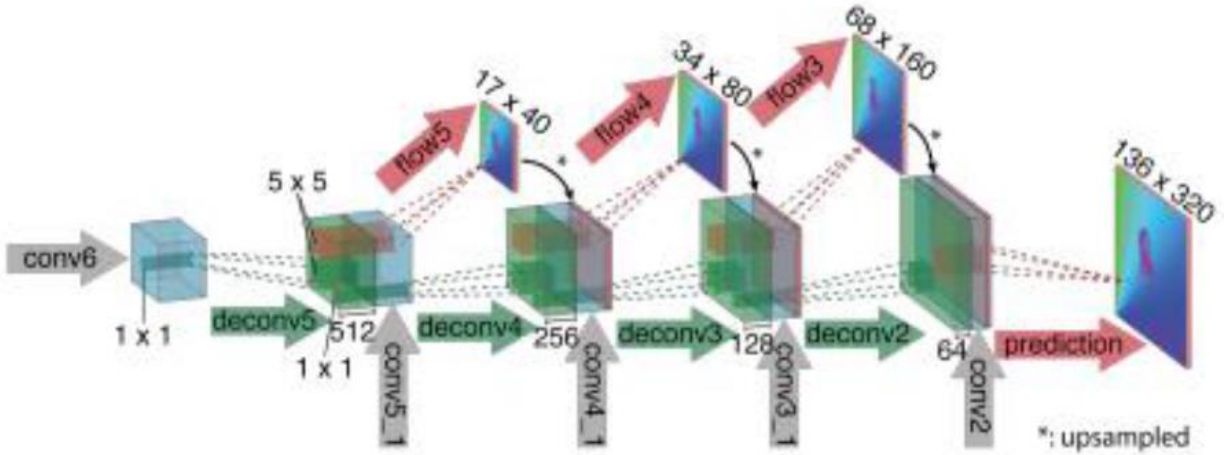
FlowNetCorr

- First process the images separately, and then correlate their features at different locations and process further.



Refinement

- Makes use of up-convolutional layers and features from contracting part



Flying Chairs

- Existing datasets are too small to train the CNN
- To provide enough training images, they generate a synthetic dataset called Flying Chairs.
- Collected images from Flicker and available rendered set of 3D chair model.
- Add images of multiple Chair to the background

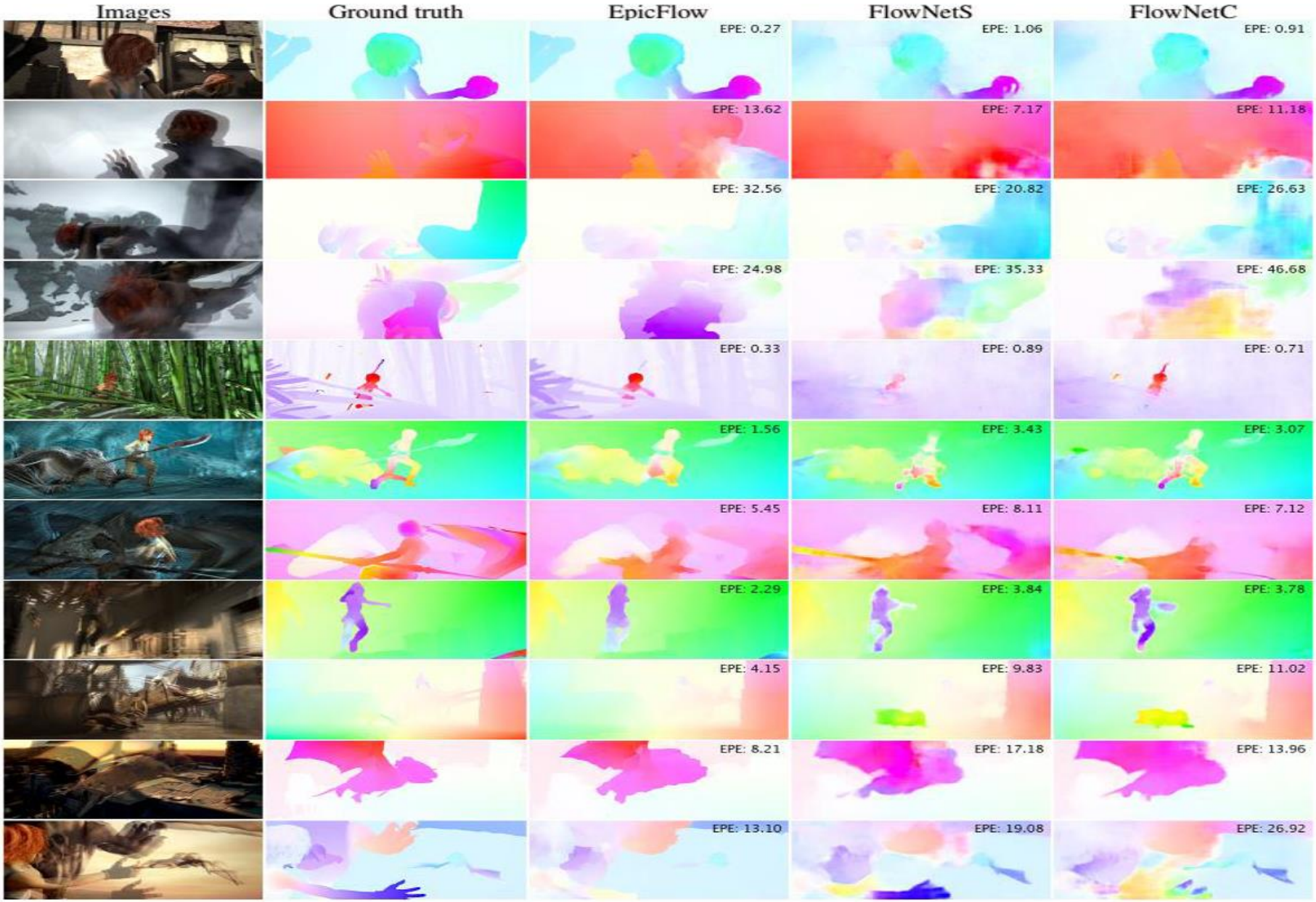
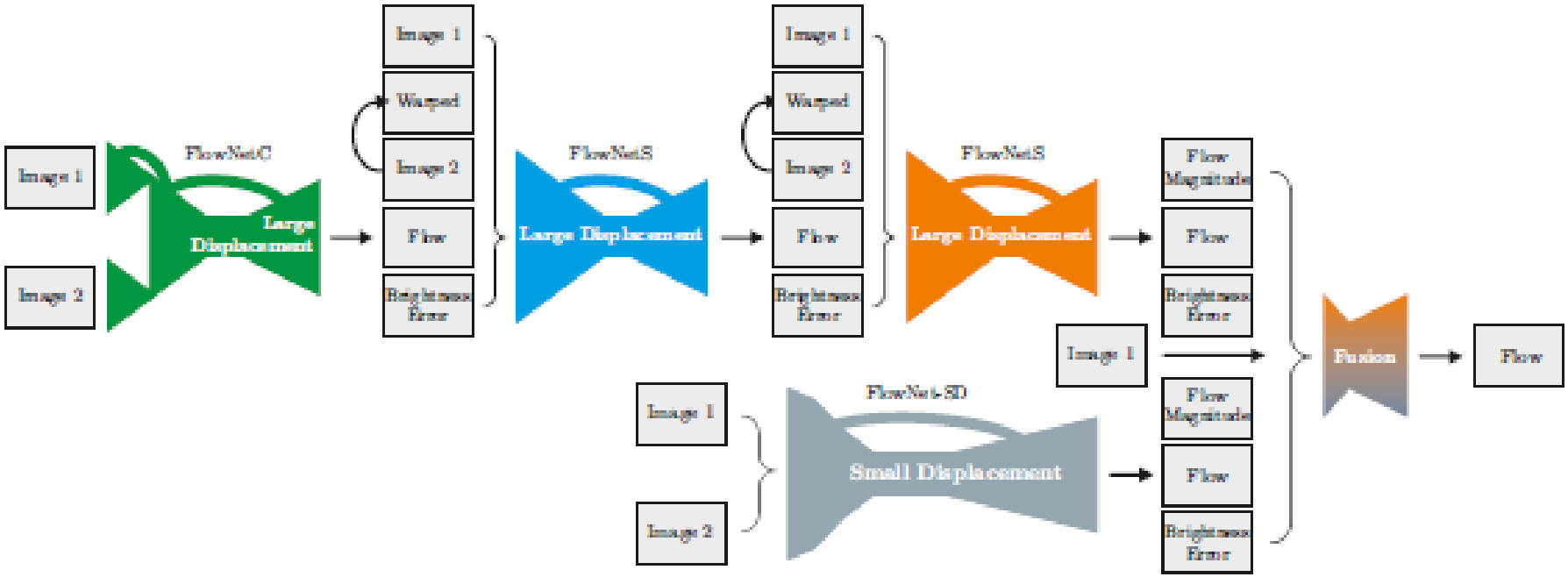


Figure 7. Examples of optical flow prediction on the Sintel dataset. In each row left to right: overlaid image pair, ground truth flow and 3 predictions: EpicFlow, FlowNetS and FlowNetC. Endpoint error is shown for every frame. Note that even though the EPE of FlowNets is usually worse than that of EpicFlow, the networks often better preserve fine details.

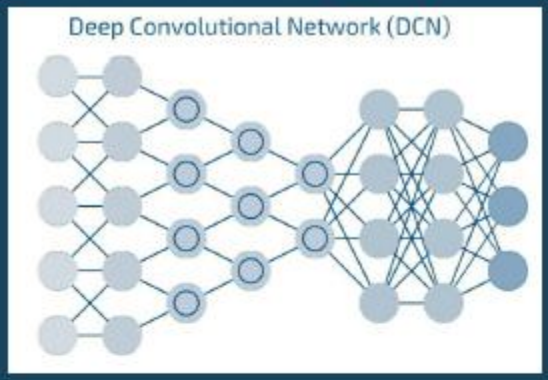
FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks



https://youtu.be/JSzUdVBmQP4

SELF-SUPERVISION

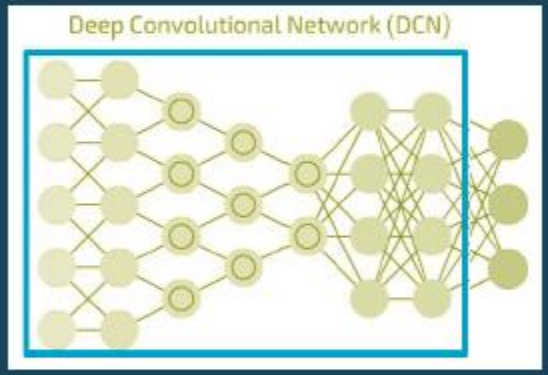
Strong Supervision



Monarch Butterfly
Race Car
Sandwich
Picnic
Train

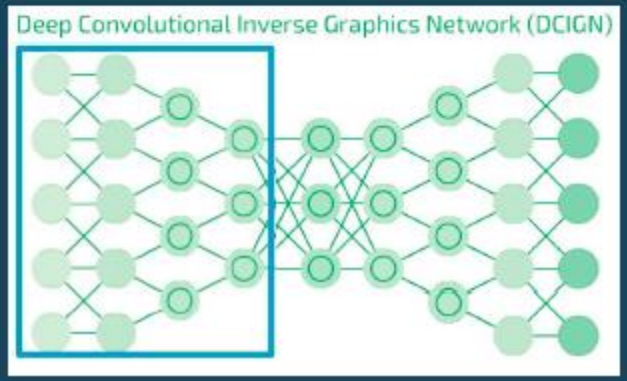
Weak Supervision

"I saw a butterfly during picnic"



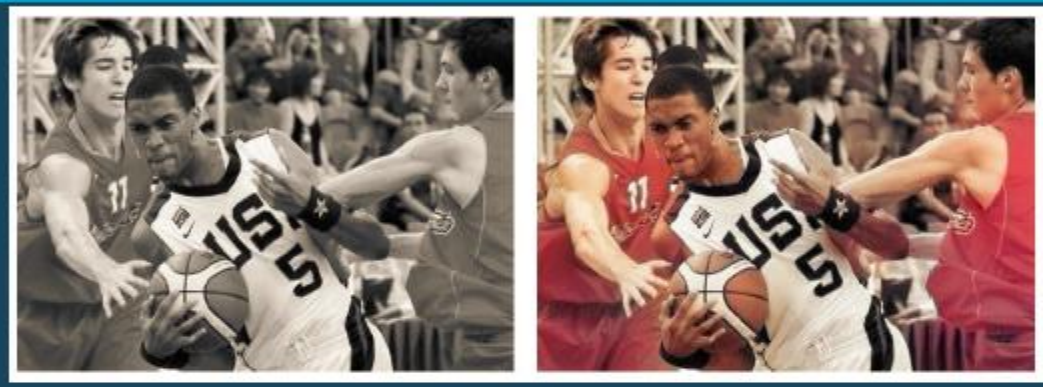
Butterfly
Race Car
Sandwich
Picnic
Train

Self-Supervision



EXAMPLES OF SELF-SUPERVISION

I Context

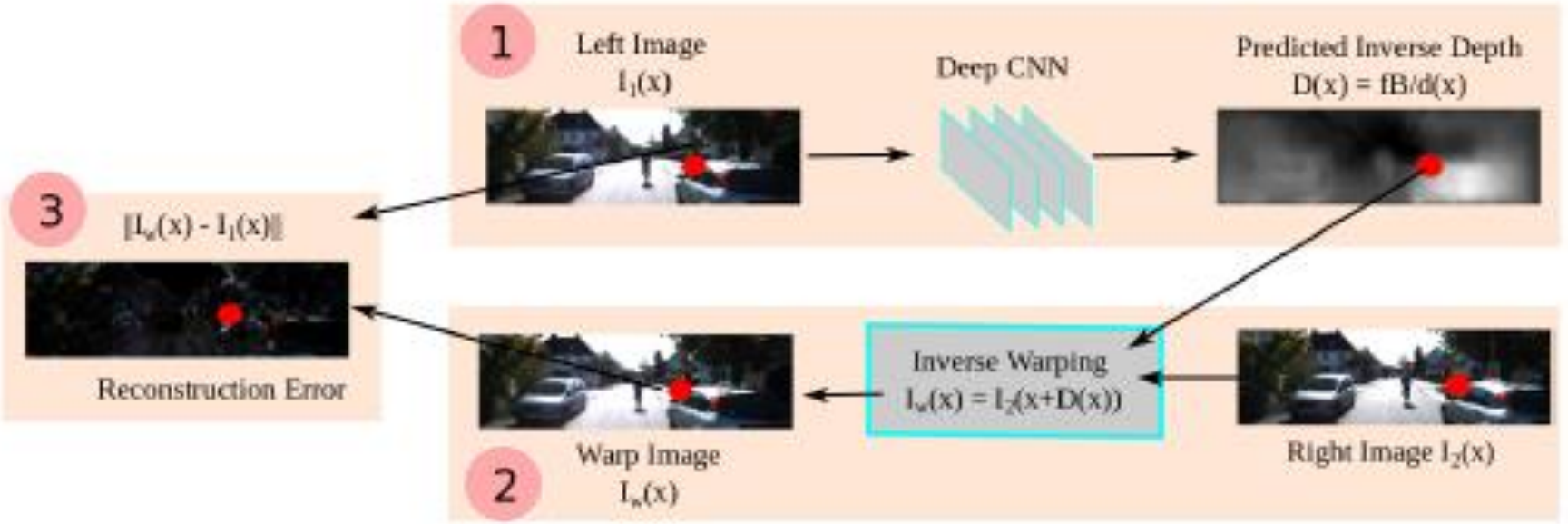


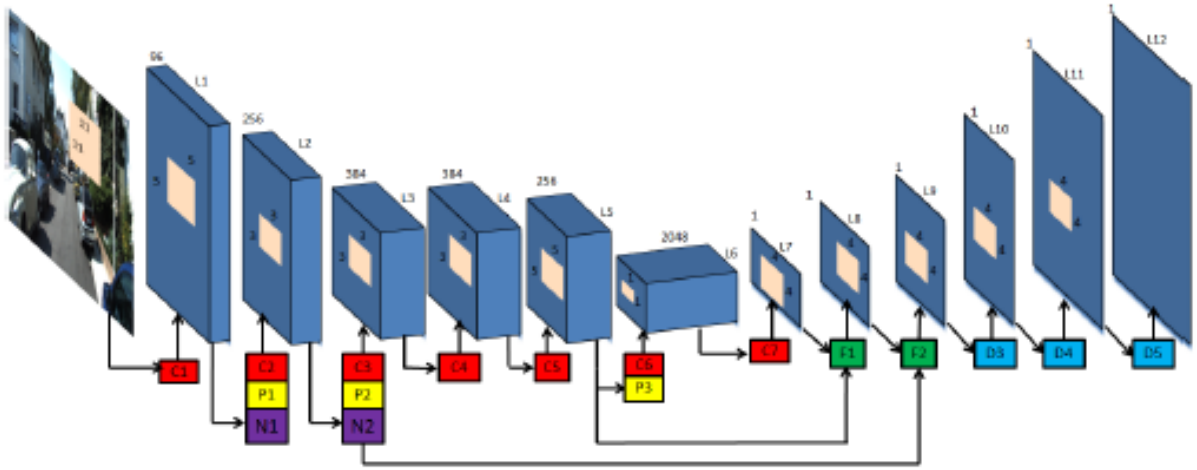
II Color

III Motion

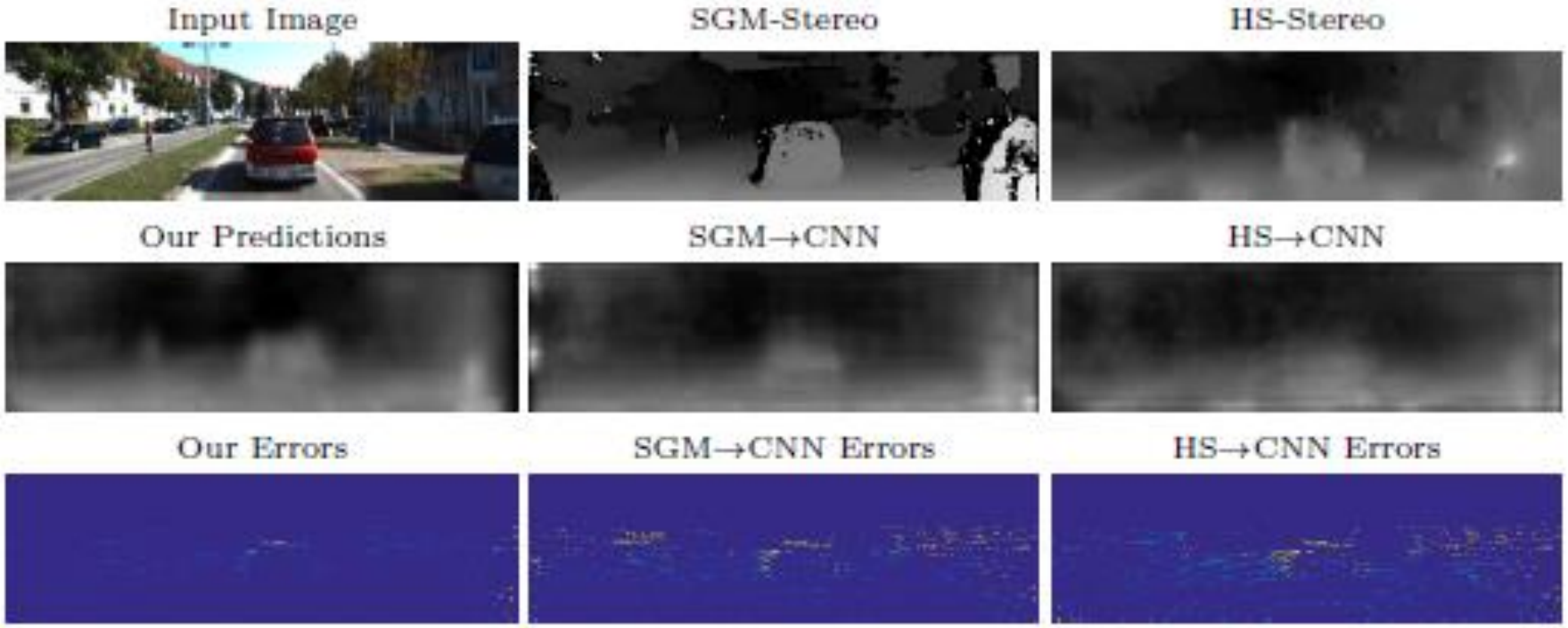
IV Ambient Noise & Noisy Labels

Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue





Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue



Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness