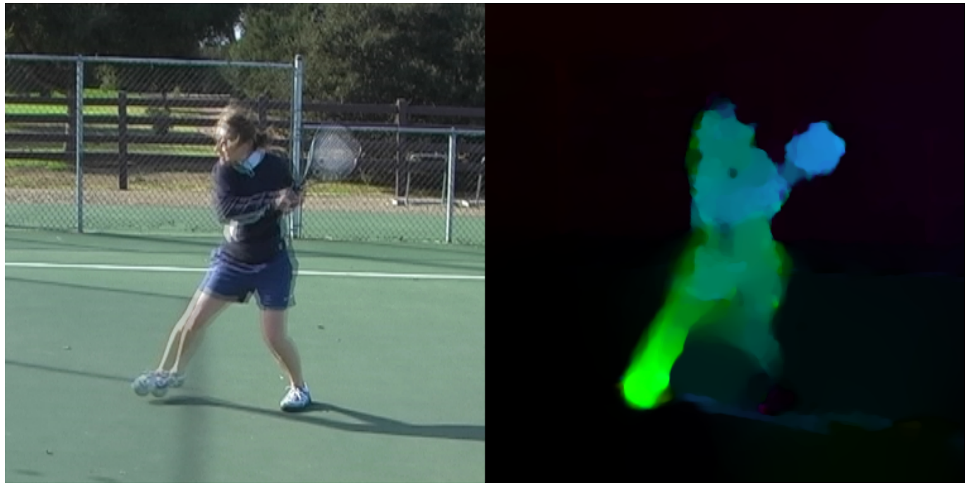
DenseNet for Dense Flow

Yi Zhu   Shawn Newsam
University of California, Merced
{yzhu25, snewsam}@ucmerced.edu

Dense Optical Flow Estimation Problem

- Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene.



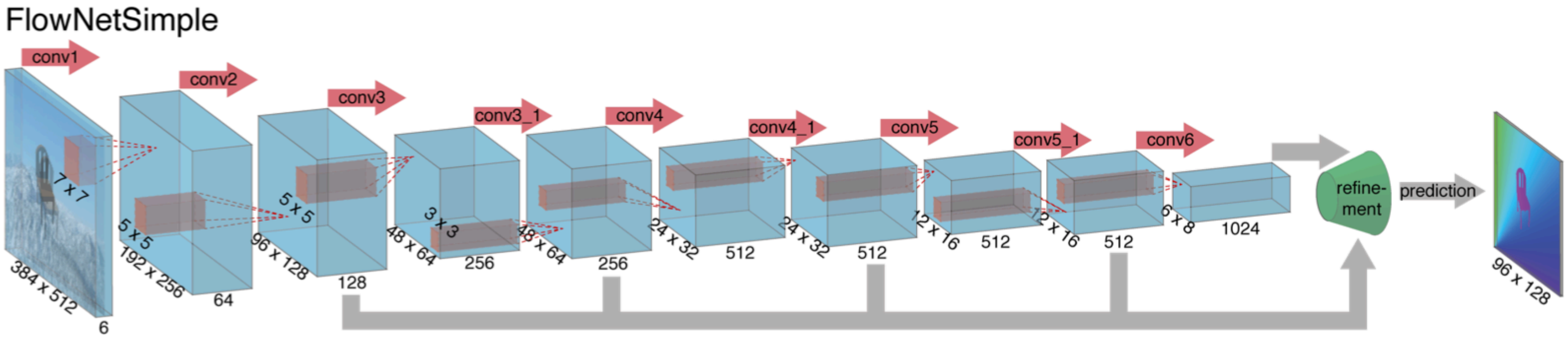
Motivation

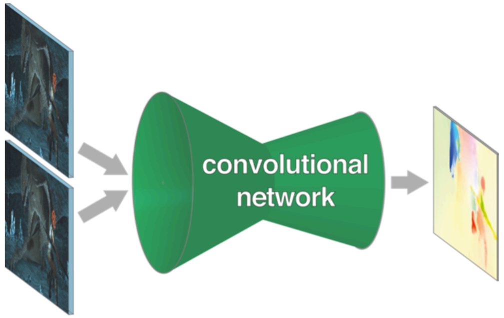
- Optical flow is useful for many vision applications, such as video object segmentation, human activity recognition, video stabilization, video tracking, etc. Specifically, scene flow for **autonomous driving** and **3D gaming**.

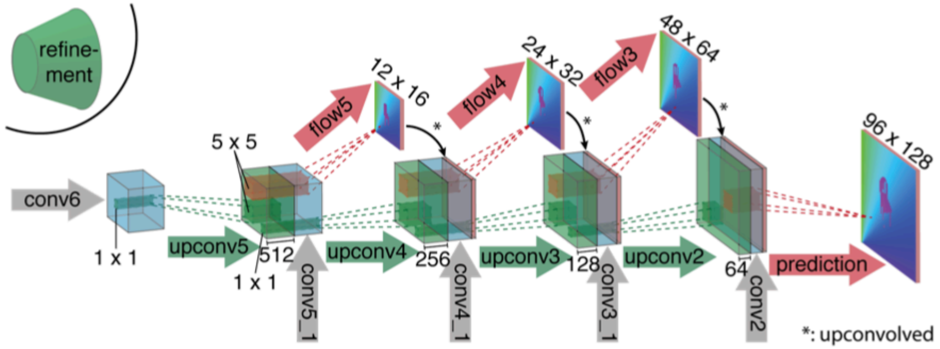
- Classical methods for estimating optical flow is often based on a variational model and solved as an energy minimization process, which is too slow for real-time applications.

- Recent CNN based approaches adopt one basic architecture: FlowNet, which may not be the optimal architecture for dense per-pixel estimation problem.

FlowNet







Alexey Dosovitskiy, et al. FlowNet: Learning Optical Flow with Convolutional Networks, in ICCV 2015

Recent literature

- A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation, CVPR 2016
- Unsupervised Convolutional Neural Networks for Motion Estimation, ICIP 2016
- Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness, ECCVW 2016
- Guided Optical Flow Learning, CVPRW 2017
- Unsupervised Monocular Depth Estimation with Left-Right Consistency, CVPR 2017
- Hidden Two-Stream Convolutional Networks for Action Recognition, arxiv 2017
- FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks, CVPR 2017
- ……………………

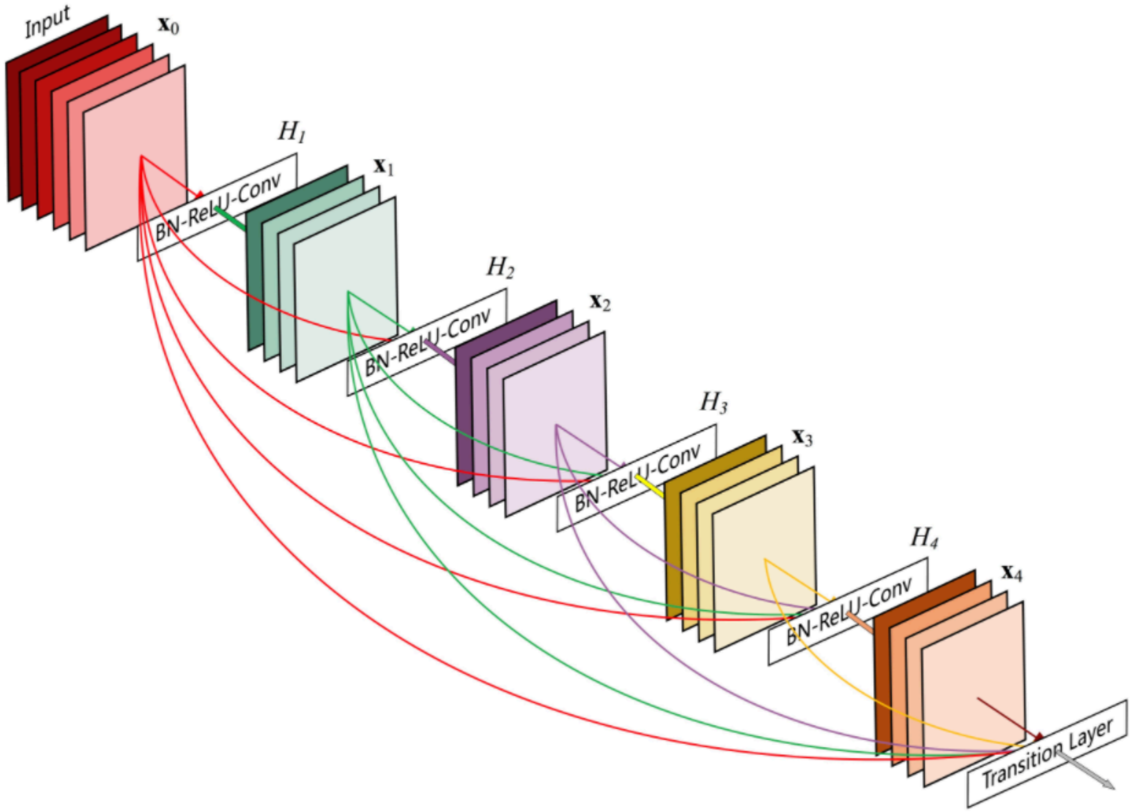
Many work use such architecture, which is also known as U-net. However, this architecture only use basic forward CNN without any fancy internal connection pattern. Could we do better?

Proposed Approach

- We propose to use DenseNet. This specific architecture is ideal for the problem at hand as it provides shortcut connections throughout the network, which leads to implicit deep supervision.

- We treat the optical flow estimation as an image reconstruction problem, which turns it to a unsupervised learning paradigm. This is ideal because it is difficult to build a large-scale dataset with ground truth optical flow.

DenseNet



- Heavy feature reuse. Model is more compact and less prone to overfitting.
- Keep high frequency image details until the end of the network.
- Each individual layer receives direct supervision from the loss function through the shortcut paths, which provides implicit deep supervision.

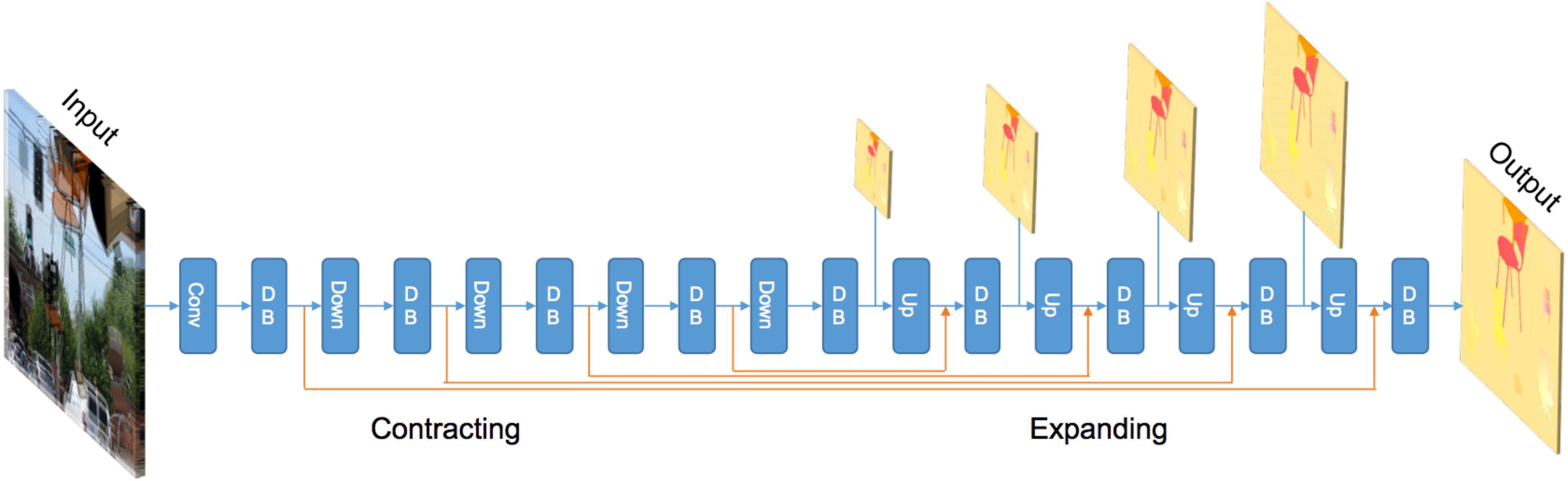
Gao Huang, et al. Densely Connected Convolutional Networks, in CVPR 2017

Fully Convolutional DenseNet

Layer
Batch Normalization
ReLU
3×3 Convolution
Dropout $p = 0.2$

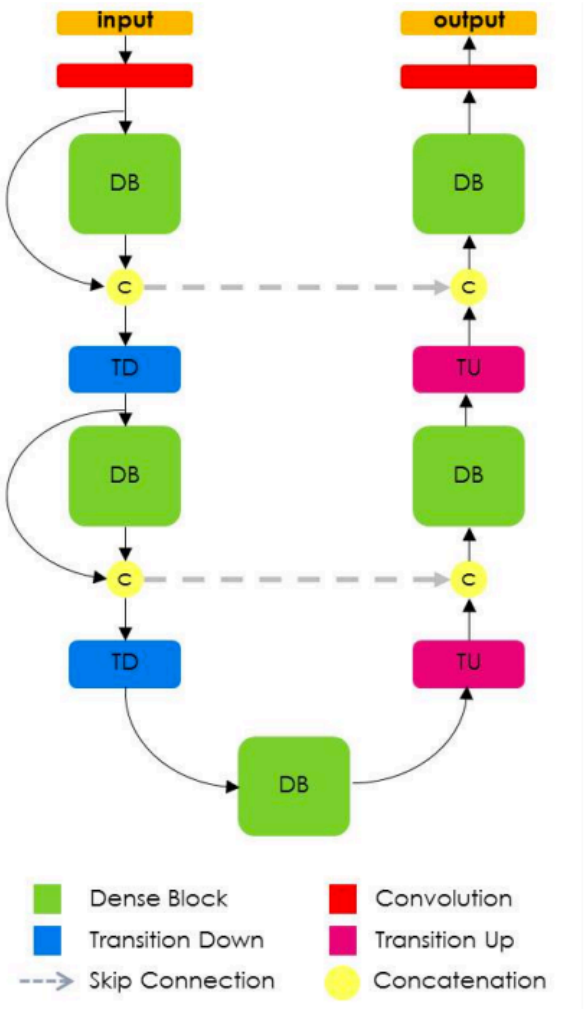
Transition Down (TD)
Batch Normalization
ReLU
1×1 Convolution
Dropout $p = 0.2$
2×2 Max Pooling

Transition Up (TU)
3×3 Transposed Convolution $stride = 2$



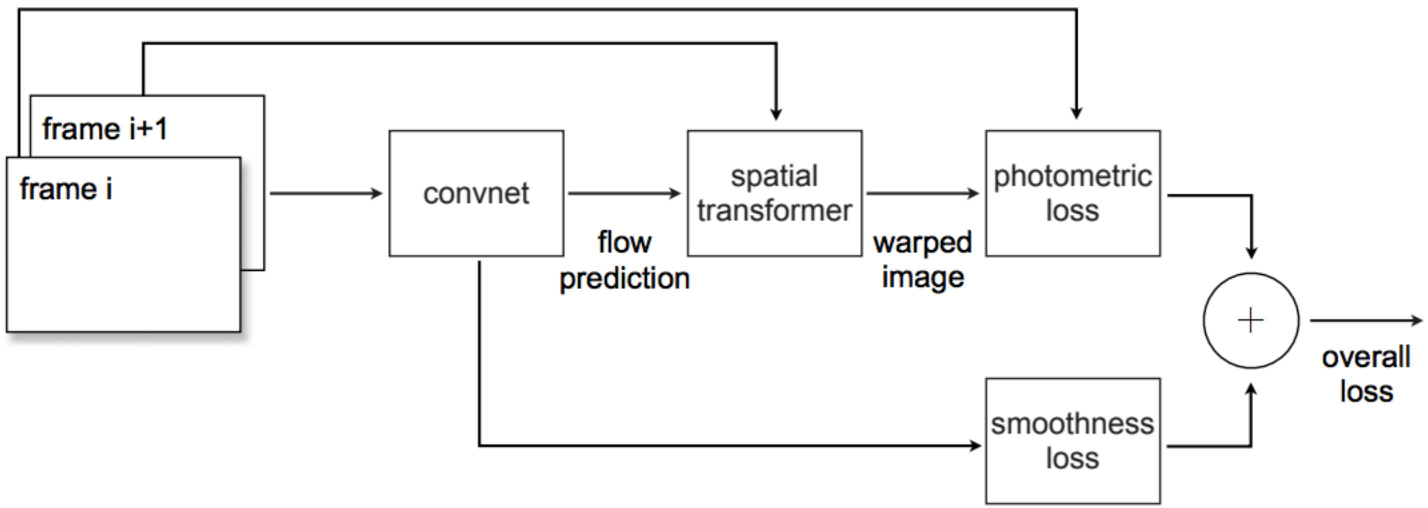
Upsampling

- Memory demanding: both feature channels and feature map resolution are increasing
- No input concatenation during upsampling path



S Jégou et al., The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation, arXiv:1611.09326, 2016

Unsupervised Learning



$$\ell_{\text{photometric}}(\mathbf{u}, \mathbf{v}; I(x,y,t), I(x,y,t+1)) = \sum_{i,j} \rho_D(I(i,j,t) - I(i + u_{i,j}, j + v_{i,j}, t+1)),$$

$$\ell_{\text{smoothness}}(\mathbf{u}, \mathbf{v}) = \sum_{j}^{H} \sum_{i}^{W} [\rho_S(u_{i,j} - u_{i+1,j}) + \rho_S(u_{i,j} - u_{i,j+1}) + \rho_S(v_{i,j} - v_{i+1,j}) + \rho_S(v_{i,j} - v_{i,j+1})],$$

Jason Yu, et al., Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness, ECCVW 2016

Quantitative Results

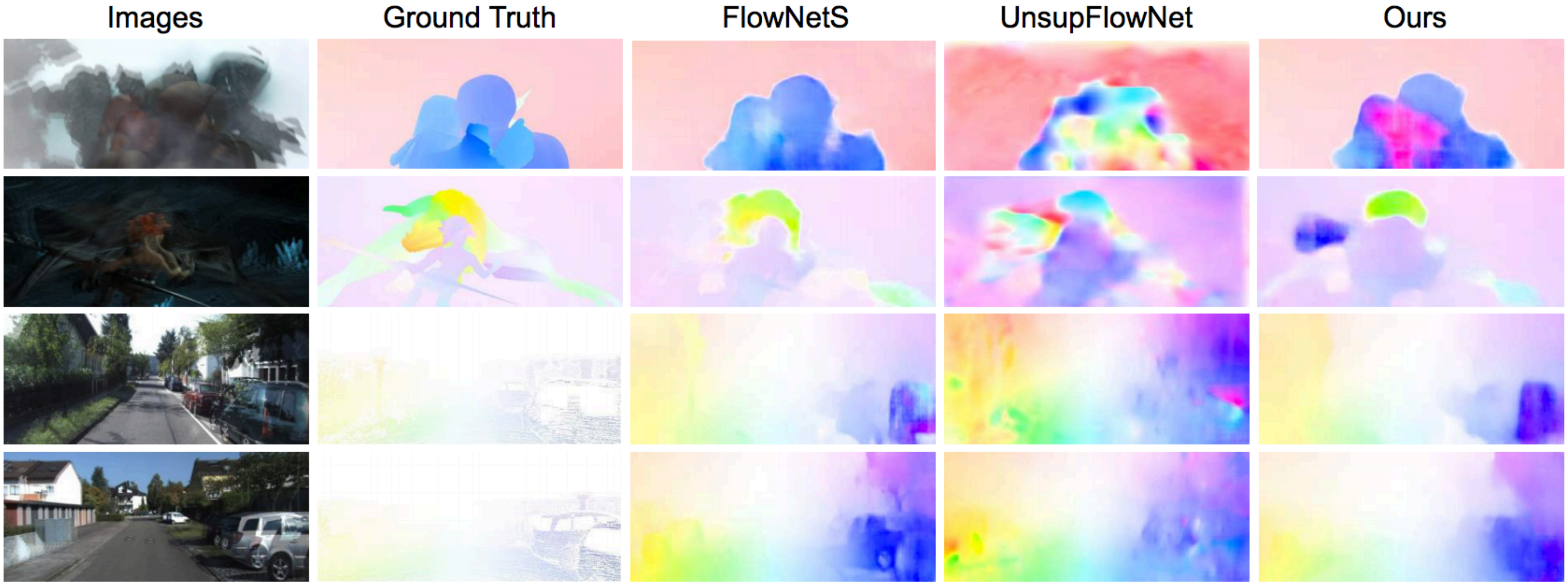
Method	Chairs	Sintel	KITTI
UnsupFlowNet [6]	5.30	11.19	12.4
VGG16 [13]	5.47	11.35	12.7
ResNet18 [14]	5.22	10.98	12.3
DenseNet [12]	5.01	10.66	12.1
DenseNet + Dense Upsampling	**4.73**	**10.07**	**11.6**
DenseNet + Dense Upsampling (Deeper)	6.65	13.46	14.0

Table 1. Optical flow estimation results on the test set of Chairs, Sintel and KITTI. All performances are reported using average EPE, lower is better. Top: Comparison of different architectures with classical upsampling. Bottom: Our proposed DenseNet with dense block upsampling.

Method	Chairs	Sintel	KITTI	Runtime
EPPM [21]	–	8.38	9.2	0.25
PCA-Flow [22]	–	8.65	6.2	0.19*
DIS-Fast [23]	–	10.13	14.4	0.02*
FlowNetS [1]	2.71	8.43	9.1	0.06
USCNN [5]	–	8.88	–	–
UnsupFlowNet [6]	5.30	11.19	12.4	0.06
Ours	4.73	10.07	11.6	0.13

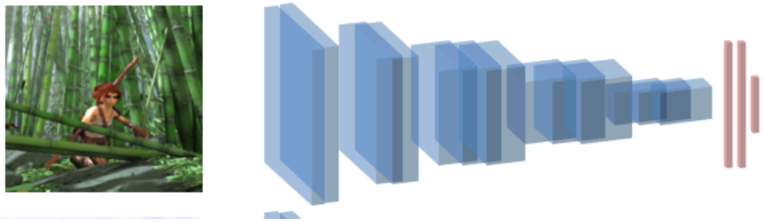
Table 2. State-of-the-art comparison. Runtime is reported in seconds per frame. Top: Classical approaches. Bottom: CNN-based approaches. * indicates the algorithm is evaluated using CPU, while the rest are on GPU.

Visual Samples

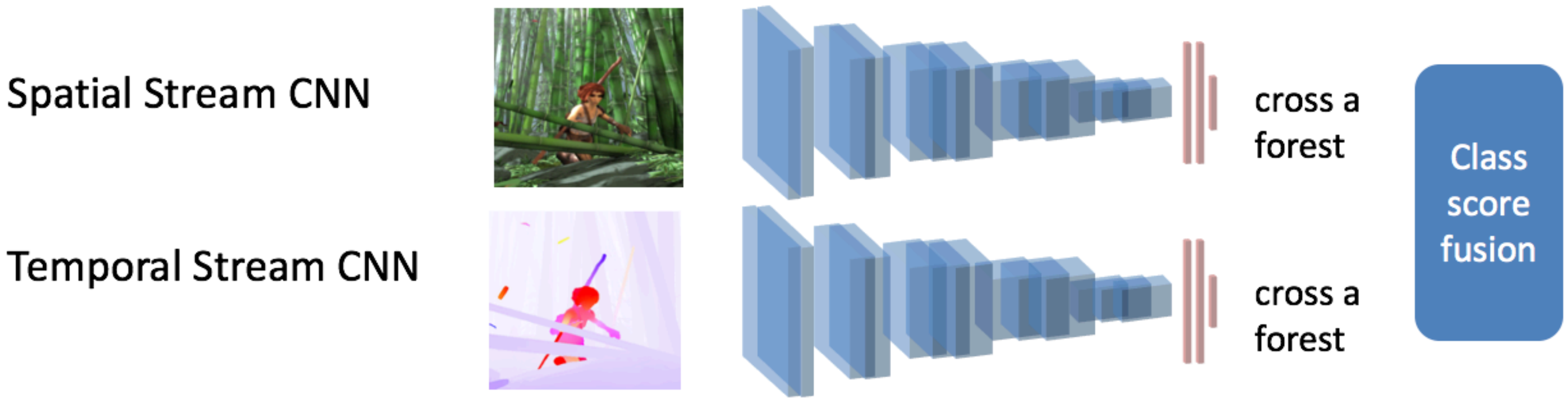


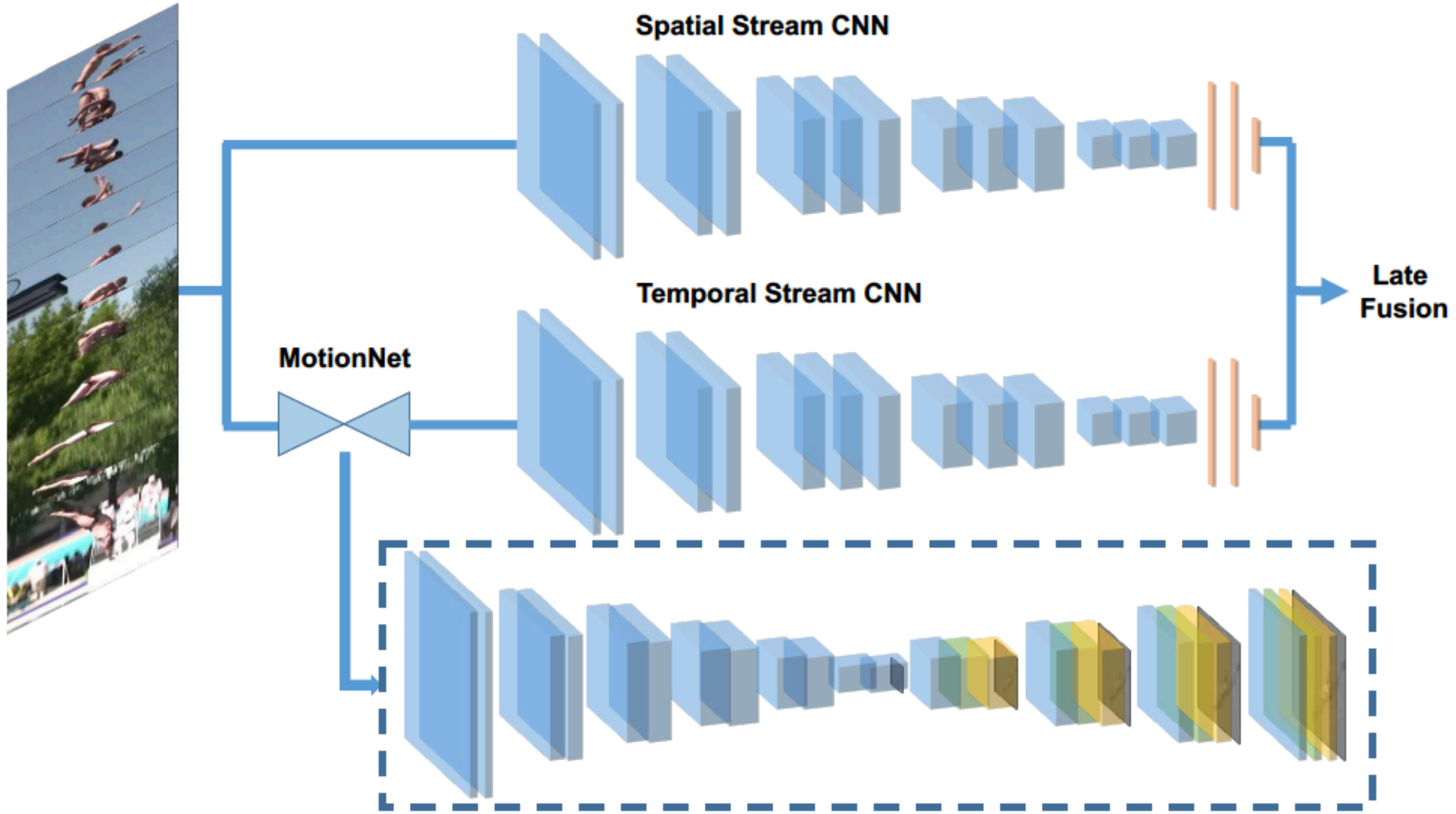
Use Flow for Action Recognition

Spatial Stream CNN



cross a
forest


Spatial Stream CNN
Temporal Stream CNN
cross a forest
cross a forest
Class score fusion



https://github.com/bryanyzhu/Hidden-Two-Stream

Method	Accuracy (%)	fps
TV-L1 [25]	85.65	14.75
FlowNet [21]	55.27	52.08
FlowNet2 [32]	79.64	8.05
NextFlow [48]	72.2	42.02
Enhanced Motion Vectors [31]	79.3	390.7
MotionNet (2 frames)	84.09	48.54
ActionFlowNet (2 frames)[18]	70.0	200.0
ActionFlowNet (16 frames)[18]	83.9	–
Stacked Temporal Stream CNN (a)	83.76	169.49
Stacked Temporal Stream CNN (b)	84.04	169.49
Stacked Temporal Stream CNN (c)	84.88	169.49
Two-Stream CNNs [10]	88.0	14.3
Very Deep Two-Stream CNNs[11]	**90.9**	**12.8**
Hidden Two-Stream CNNs (a)	87.50	120.48
Hidden Two-Stream CNNs (b)	87.99	120.48
Hidden Two-Stream CNNs (c)	**89.82**	**120.48**

Method	UCF101(%)	HMDB51(%)
Motion Vector + Fisher Vector Encoding [58]	78.5	46.7
ActionFlowNet (2 frames) [18]	70.0	42.6
ActionFlowNet (16 frames) [18]	83.9	56.4
C3D (1 Net) [6]	82.3	–
C3D (3 Net) [6]	85.2	–
Enhanced Motion Vector [31]	80.2	–
RGB + Enhanced Motion Vector [31]	86.4	–
RGB Diff [15]	83.0	–
RGB + RGB Diff [15]	86.8	–
Two-Stream 3DNet Initial [57]	85.2	–
Two-Stream 3DNet Mid [57]	87.0	–
Hidden Two-Stream Networks with Tiny-MotionNet	88.7	58.9
Hidden Two-Stream Networks with MotionNet	**90.3**	**60.5**

Conclusion

- We extend the current **DenseNet** architecture to a fully convolutional network.
  - Due to the dense connectivity pattern, our proposed method achieves better flow accuracy than the previous best unsupervised approach and shortens the performance gap with supervised ones.

- We use image reconstruction loss as guidance to learn motion estimation in an **unsupervised** learning manner.
  - Due to unsupervised learning, we can experiment with large-scale video corpora in future work, to learn non-rigid real world motion patterns.

Acknowledgement: This work was funded in part by a National Science Foundation CAREER grant, #IIS-1150115, and a seed grant from the Center for Information Technology in the Interest of Society (CITRIS). We gratefully acknowledge the support of NVIDIA Corporation through the donation of the Titan X GPU used in this work.

Q&A

Please come to our poster for more details. Thank you.