DENSENET FOR DENSE FLOW

Yi Zhu and Shawn Newsam

University of California, Merced
5200 N Lake Rd, Merced, CA, US
{yzhu25, snewsam}@ucmerced.edu

ABSTRACT

Classical approaches for estimating optical flow have achieved rapid progress in the last decade. However, most of them are too slow to be applied in real-time video analysis. Due to the great success of deep learning, recent work has focused on using CNNs to solve such dense prediction problems. In this paper, we investigate a new deep architecture, Densely Connected Convolutional Networks (DenseNet), to learn optical flow. 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 extend current DenseNet to a fully convolutional network to learn motion estimation in an unsupervised manner. Evaluation results on three standard benchmarks demonstrate that DenseNet is a better fit than other widely adopted CNN architectures for optical flow estimation.

***Index Terms*—** Optical flow estimation, Unsupervised learning, Convolutional neural network

1. INTRODUCTION

Convolutional Neural Networks (CNNs), due to their immense learning capacity and superior efficiency, have advanced a variety of computer vision tasks, including optical flow prediction. Recent work [1, 2] built large-scale synthetic datasets to train a supervised CNN and show that networks trained on such unrealistic data still generalize very well to existing datasets such as Sintel [3] and KITTI [4]. Some other work [5, 6, 7] designed new objectives like image reconstruction loss to guide the network learning in an unsupervised way for motion estimation. Though [1, 2, 5, 6] are totally different approaches, they all use variants of one architecture, the "FlowNet Simple" network [1].

FlowNetS is a conventional CNN architecture, consisting of a contracting part and an expanding part. Given adjacent frames as input, the contracting part uses a series of convolutional layers to extract high level semantic features, while the expanding part tries to predict the optical flow at the original image resolution by successive deconvolutions. In between, it uses skip connections [8] to provide fine image details from lower layer feature maps. This generic pipeline, *contract, expand, skip connections*, is widely adopted for per-pixel prediction problems, such as semantic segmentation [9], depth estimation [10], video coloring [11], etc.

However, skip connection is a simple strategy towards combining coarse semantic features and fine image details; it is not involved in the learning process. What we desire is to keep the high frequency image details till the end of the network in order to provide implicit deep supervision. Simply put, we want to ensure maximum information flow between layers in the network.

DenseNet [12], a recently proposed CNN architecture, has an interesting connectivity pattern: every layer is connected to each other within a dense block. In this case, all layers can access feature maps from their preceding layers which encourage heavy feature reuse. As a direct consequence, the model is more compact and less prone to overfitting. Besides, each individual layer receives direct supervision from the loss function through the shortcut paths, which provides implicit deep supervision. All these good properties make DenseNet a natural fit for per-pixel prediction problems. There is a concurrent work using DenseNet for semantic segmentation [9], which achieved state-of-the-art performance without neither pretraining nor additional post-processing. However, estimating optical flow is different from semantic segmentation. We will illustrate the differences in Section 3.

In this paper, we propose to use DenseNet for optical flow prediction. Our contributions are two-fold. First, we extend current DenseNet to a fully convolutional network. Our model is totally unsupervised, and achieves performance close to supervised approaches. Second, we empirically show that replacing convolutions with dense blocks in the expanding part yields better performance.

2. METHOD

Given adjacent frames, previous I_1 and next I_2, our goal is to learn a model that can predict per-pixel motion field (U, V) between the two images. U and V are the horizontal and vertical displacement. In this section, we first review the DenseNet architecture, and then outline our unsupervised learning framework based on a fully convolutional DenseNet.

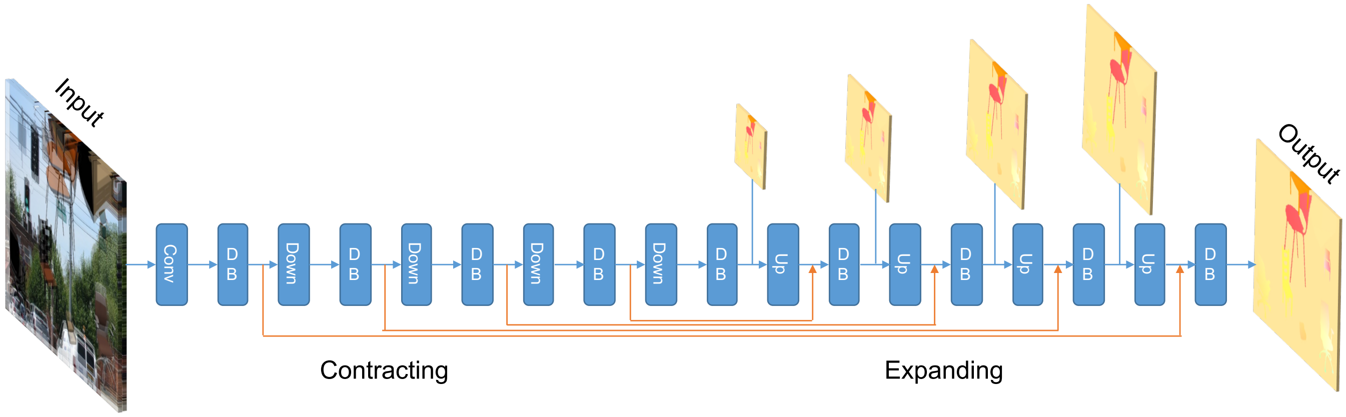

Fig. 1. An overview of our unsupervised learning framework based on dense blocks (DB). “Down” is the transition down layer, and “Up” is the transition up layer. The orange colored arrows indicate the skip connections. See more details in Section 2.2.

2.1. DenseNet Review

Traditional CNNs, such as FlowNetS, calculate the output of l^{th} layer by applying nonlinear transformation H to the previous layer's output x_{l-1},

$$x_l = H_l(x_{l-1}). \tag{1}$$

Through consecutive convolution and pooling, the network will achieve spatial invariance and obtain coarse semantic features in the top layers. However, fine image details tend to disappear in the very top of the network.

To improve information flow between layers, DenseNet [12] provides a simple connectivity pattern: the l^{th} layer receives the features maps of all preceding layers as inputs:

$$x_l = H_l([x_0, x_1, ..., x_{l-1}]) \tag{2}$$

where $[x_0, x_1, ..., x_{l-1}]$ is a single tensor constructed by concatenation of the previous layers' output feature maps. In this manner, even the last layer can have access to the input information of the first layer. And all layers receive direct supervision from the loss function through the shortcut connections. $H_l(\cdot)$ is a composite function of four consecutive operations, batch normalization (BN), leaky rectified linear units (LReLU), a 3×3 convolution and dropout. We denote such composite function as one layer.

In our experiments, the DenseNet in the contracting part has four dense blocks, each of which has four layers. Between the dense blocks, there are transition down layers consisting of a 1×1 convolution followed by a 2×2 max pooling. We compare it with three other popular architectures, namely FlowNetS [1], VGG16[13] and ResNet18 [14] in Section 3.3.

2.2. Fully Convolutional DenseNet

Classical expanding uses series of convolution, deconvolution, and skip connections to recover the spatial resolution in order to get the per-pixel prediction results. Due to the good properties of DenseNet, we propose to replace the convolutions with dense blocks as well during expanding.

However, if we follow the same dense connectivity pattern, the number of feature maps after each dense block will keep increasing. Considering the resolution of feature maps also increase during expanding, the computational cost will be intractable for current GPUs. Thus, for a dense block in the expanding part, we do not concatenate input to its final output. For example, if the input has k_0 channels, the output of an L layer dense block will have a number of Lk feature maps. k is the growth rate of a DenseNet, defining the number of feature maps each layer produces. Note that dense blocks in the contracting part will output $k_0 + Lk$ feature maps.

For symmetry, we also introduce four dense blocks in the expanding part, each of which has four layers. Bottom layer feature maps at the same resolution are concatenated through skip connections. Between the dense blocks, there are transition up layers composed of two 3×3 deconvolution with a stride of 2. One is for upsampling of the estimated optical flow, the other is for upsampling of the feature maps.

2.3. Unsupervised Motion Estimation

Supervised approaches adopt synthetic datasets for CNN to learn optical flow prediction. However, synthetic motions/scenes are quite different from real world ones, thus limiting the generalizability of the learned model. Besides, even constructing synthetic dataset requires a lot of manual effort [3]. Hence, unsupervised learning is an ideal option for the naturally ill-conditioned motion estimation problem.

Recall that the unsupervised approach [6] treats the optical flow estimation as an image reconstruction problem. The intuition is that if we can use the predicted flow and the next frame to reconstruct the previous frame, our network is learning useful representations about the underlying motions. To be specific, we denote the reconstructed previous frame as I'_1.

The goal is to minimize the photometric error between the previous frame I_1 and the inverse warped next frame I_1':

$$L_{\text{reconst}} = \frac{1}{N}\sum_{i,j}^{N} \rho(I_1(i,j) - I_1'(i,j)). \tag{3}$$

Here $I_1'(i,j) = I_2(i + U_{i,j}, j + V_{i,j})$. N is the total number of pixels. The inverse warp is done by using spatial transformer modules [15] inside the CNN. We use a robust convex error function, the generalized Charbonnier penalty $\rho(x) = (x^2 + \epsilon^2)^\alpha$, to reduce the influence of outliers. This reconstruction loss is similar to the brightness constancy objective in classical variational formulations.

An overview of our unsupervised learning framework based on DenseNet is illustrated in Fig. 1. Our network has a total of 53 layers with a growth rate of 12. But due to parameter efficiency of dense connectivity, our model only has 2M parameters, while FlowNetS has 38M.

3. EXPERIMENTS

3.1. Datasets

Flying Chairs [1] is a synthetic dataset designed specifically for training CNNs to estimate optical flow. It is created by applying affine transformations to real images and synthetically rendered chairs. The dataset contains 22,872 image pairs: 22,232 training and 640 test samples according to the standard evaluation split.

MPI Sintel [3] is also a synthetic dataset derived from a short open source animated 3D movie. There are 1,628 frames, 1,064 for training and 564 for testing. In this work, we only report performance on its final pass because it contains sufficiently realistic scenes including natural image degradations.

KITTI Optical Flow 2012 [4] is a real world dataset collected from a driving platform. It consists of 194 training image pairs and 195 test pairs with sparse ground truth flow. We report the average endpoint error (EPE) on the entire image.

Since the dataset size of Sintel/KITTI is relatively small, we first pretrain our network on Chairs, and then fine tune it to report their performance. Note that the fine tuning here is also unsupervised, we are not using ground truth flow from Sintel/KITTI.

3.2. Implementation

During unsupervised training, we calculate reconstruction loss for each expansion. There are 5 expansions in our network, resulting in 5 losses. We use the same loss weights as in [6]. The generalized Charbonnier parameter α is 0.25 and ϵ is 0.001. The models are trained using Adam optimization with default parameters, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate is set to 10^{-5}, and then divided by half every 100k. We end our training at 600k iterations. We apply the same data augmentations as in [6] to prevent overfitting.

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.

3.3. Results and Discussion

We have three observations given the results in Table 1.

Observation 1: As shown in top section of Table 1, all four popular architectures perform reasonably well on optical flow prediction. The reason why VGG16 performs the worst is that multiple pooling layers may lose the image details. On the contrary, ResNet18 only has one pooling layer in the beginning, so it performs better than both VGG16 and FlowNetS. Interestingly, DenseNet also has multiple pooling layers, but due to dense connectivity, we don't lose fine appearance information. Thus, as expected, DenseNet performs the best with the least number of parameters.

Inspired by success of using deeper models, we also implement a network with five dense blocks in both the contracting and expanding parts, where each block has ten layers. However, as shown in the last row of Table 1, the performance is much worse due to overfitting. This may indicate that optical flow is a low-level vision problem, that doesn't need a substantially deeper network to achieve better performance.

Observation 2: Using dense blocks during expanding is beneficial. In Table 1, DenseNet with dense upsampling achieves better performance on all three benchmarks than DenseNet with classical upsampling, especially on Sintel. As Sintel has much more complex context than Chairs and KITTI, it may benefit more from the implicit deep supervision. This confirms that using dense blocks instead of a single convolution can maintain more information during the expanding process, which leads to better flow estimates.

Observation 3: One of the advantages of DenseNet is that it is less prone to overfitting. The authors in [12] have shown that it can perform well even when there is no data augmentation compared to other network architectures. We investigate this by directly training from scratch on Sintel, without pretraining using Chairs. We built the training dataset using image pairs from both the final and clean passes of Sintel. When we use the same implementation and training strategies, the flow estimation performance is 10.3, which is very close to 10.07. One possible reason for such robustness is because of the model compactness and implicit deep supervision provided by DenseNet. This is ideal for optical flow estimation since most benchmarks have limited training data.

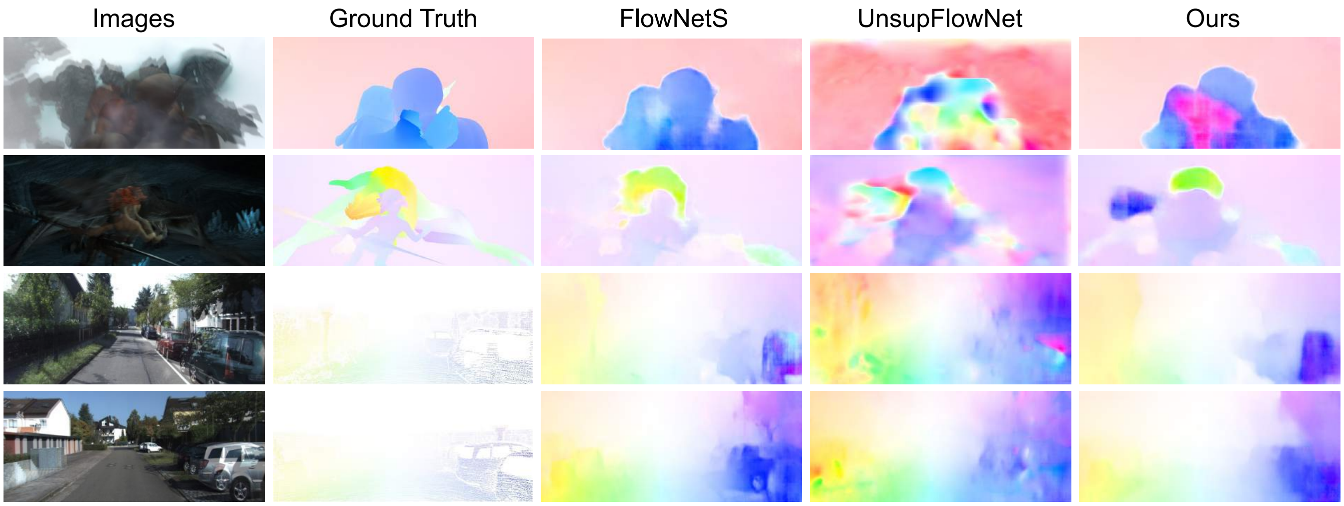

Fig. 2. Visual examples of predicted optical flow from different methods. Top two are from Sintel, and bottom two from KITTI.

3.4. Comparison to State-of-the-arts

In this section, we compare our proposed method to recent state-of-the-art approaches. We only consider approaches that are fast because optical flow is often used in time sensitive applications. We evaluated all CNN-based approaches on a workstation with Intel Core I7 with 4.00GHz and a Nvidia Titan X GPU. For classical approaches, we just use their reported runtime.

As shown in Table 2, although unsupervised learning still lags behind supervised approaches [1], our network based on fully convolutional DenseNet shortens the performance gap and achieves lower EPE on three standard benchmarks than state-of-the-art unsupervised approach [6]. Comparing to [5], we get higher EPE on Sintel because they use variational refinement technique.

We also show some visual examples in Figure 2. We can see that supervised FlowNetS can estimate optical flow close to ground truth, while UnsupFlowNet struggles to maintain fine image details and generates very noisy flow estimation. Due to dense connectivity pattern, our proposed method can produce much smoothier flow than UnsupFlowNet, and recover the high frequency image details, such as human boundaries and car shapes.

Therefore, we demonstrate that DenseNet is a better fit for dense optical flow prediction, both quantitatively and qualitatively. However, by exploring different network architectures, we found that existing networks perform similarly on predicting optical flow. We may need to design new operators like the correlation layer in [1], or novel architectures [16, 17] to learn motions between adjacent frames in future work. The model should handle both large and small displacement, as well as fine motion boundaries. Another concern of this work is that DenseNet has a large memory bandwidth which may limit its potential applications.

Method	Chairs	Sintel	KITTI	Runtime
EPPM [18]	–	8.38	9.2	0.25
PCA-Flow [19]	–	8.65	6.2	0.19*
DIS-Fast [20]	–	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.

4. CONCLUSION

In this paper, we extend the current DenseNet architecture to a fully convolutional network, and use image reconstruction loss as guidance to learn motion estimation. Due to the dense connectivity pattern, our proposed method achieves better flow accuracy than previous best unsupervised approach [6], and shortens the performance gap to supervised ones. Besides, our model is totally unsupervised. Thus we can experiment with large-scale video corpora in future work, to learn non-rigid real world motion patterns. Through comparison of popular CNN architectures, we found that it is important to design novel operators or networks for optical flow estimation instead of relying on existing architectures for image classification.

5. ACKNOWLEDGEMENTS

This work was funded by a National Science Foundation CAREER grant, #IIS-1150115. We gratefully acknowledge the support of NVIDIA Corporation through the donation of the Titan X GPU used in this work.

6. REFERENCES

[1] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazrbas, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning Optical Flow with Convolutional Networks," in *ICCV*, 2015.

[2] Nikolaus Mayer, Eddy Ilg, Philip Husser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox, "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation," in *CVPR*, 2016.

[3] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A Naturalistic Open Source Movie for Optical Flow Evaluation," in *ECCV*, 2012.

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *CVPR*, 2012.

[5] Aria Ahmadi and Ioannis Patras, "Unsupervised Convolutional Neural Networks for Motion Estimation," in *ICIP*, 2016.

[6] Jason J. Yu, Adam W. Harley, and Konstantinos G. Derpanis, "Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness," in *ECCV Workshop*, 2016.

[7] Yi Zhu, Zhenzhong Lan, Shawn Newsam, and Alexander G. Hauptmann, "Guided Optical Flow Learning," *arXiv preprint arXiv:1702.02295*, 2017.

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully Convolutional Models for Semantic Segmentation," in *CVPR*, 2015.

[9] Simon Jgou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio, "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation," *arXiv preprint arXiv:1611.09326*, 2016.

[10] Clement Godard, Oisin Mac Aodha, and Gabriel J. Brostow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency," *arXiv preprint arXiv:1609.03677*, 2016.

[11] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Deep End2End Voxel2Voxel Prediction," in *CVPR Workshop*, 2016.

[12] Gao Huang, Zhuang Liu, Kilian Q. Weinberger, and Laurens van der Maaten, "Densely Connected Convolutional Networks," *arXiv preprint arXiv:1608.06993*, 2016.

[13] Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *ICLR*, 2015.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, 2016.

[15] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu, "Spatial Transformer Network," in *NIPS*, 2015.

[16] Anurag Ranjan and Michael J. Black., "Optical Flow Estimation using a Spatial Pyramid Network," *arXiv preprint arXiv:1611.00850*, 2016.

[17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox, "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks," *arXiv preprint arXiv:1612.01925*, 2016.

[18] L. Bao, Q. Yang, and H. Jin, "Fast Edge-Preserving PatchMatch for Large Displacement Optical Flow," in *CVPR*, 2014.

[19] J. Wulff and M. Black, "Efficient Sparse-to-Dense Optical Flow Estimation using a Learned Basis and Layers," in *CVPR*, 2015.

[20] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool, "Fast Optical Flow using Dense Inverse Searchn," in *ECCV*, 2016.