

Hybrid Learning of Optical Flow and Next Frame Prediction to Boost Optical Flow in the Wild

Nima Sedaghat, Mohammadreza Zolfaghari, Thomas Brox
University of Freiburg
Germany

{nima, zolfagha, brox}@cs.uni-freiburg.de

Abstract

CNN-based optical flow estimation has attracted attention recently, mainly due to its impressively high frame rates. These networks perform well on synthetic datasets, but they are still far behind the classical methods in real-world videos. This is because there is no ground truth optical flow for training these networks on real data. In this paper, we boost CNN-based optical flow estimation in real scenes with the help of the freely available self-supervised task of next-frame prediction. To this end, we train the network in a hybrid way, providing it with a mixture of synthetic and real videos. With the help of a sample-variant multi-tasking architecture, the network is trained on different tasks depending on the availability of ground-truth. We also experiment with the prediction of “next-flow” instead of estimation of the current flow, which is intuitively closer to the task of next-frame prediction and yields favorable results. We demonstrate the improvement in optical flow estimation on the real-world KITTI benchmark. Additionally, we test the optical flow indirectly in an action classification scenario. As a side product of this work, we report significant improvements over state-of-the-art in the task of next-frame prediction.

1. Introduction

Supervised learning of optical flow estimation with a deep network yields a good trade-off between run time and accuracy of the estimated optical flow [6]. However, such supervised learning requires a large number of training pairs, which have been provided via synthetic images. Such imagery lacks realism and diversity, and it keeps the network from using the full potential of the learning concept. Particularly on real-world data, FlowNet [6] does not yield the same accuracy as state-of-the-art conventional optical flow estimation techniques.

In this paper, we approach this problem by providing

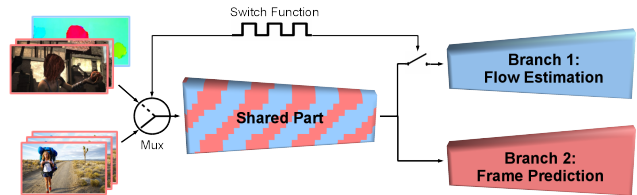


Figure 1: We improve CNN-based optical flow estimation in real videos by adding the extra *self-supervised* task of future frame prediction, and training the network with a mixture of synthetic and real-world videos. This combination is made possible by putting a “multiplexer” at the entry of the network which mixes data from the two sources on a timely basis.

real-world data to the network during training. Since there is only a very limited amount of real-world image pairs with ground truth optical flow, we use a semi-supervised hybrid multi-tasking scheme that exploits real-world videos without ground truth and synthetic imagery with ground truth. For the network to learn useful concepts from the unlabeled data, we build on the self-supervised task of next-frame prediction as an auxiliary task. The general concept of this hybrid learning task is illustrated in Figure 1.

The hybrid multi-tasking combines the best of supervised learning on synthetic data and self-supervised learning on real data. On the KITTI optical flow benchmark, we obtained a clear improvement over the FlowNet, which was trained without the self-supervised next frame prediction task. The improvement over the baseline is even larger when testing on an application task for optical flow, such as action recognition.

In addition to the hybrid multi-task learning of optical flow estimation and next frame prediction, we also propose multi-task learning on next frame prediction and next flow prediction. The latter two sub-tasks are more compatible and improve results when feeding the optical flow into an

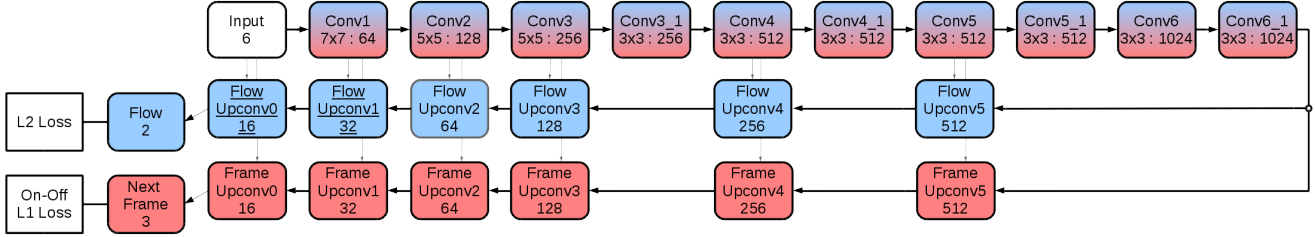


Figure 2: Details of the multi-tasking architecture. For the sake of clarity, the lower-resolution outputs and their corresponding losses (as introduced in [6]) are not displayed here. The flow estimation and frame prediction branches only differ in the number of channels of the output layer(s). Numbers in the boxes show kernel size and number of output channels for each layer. The upconvolutional layers are designed such that the output of each layer is of the same size (resolution) as its dual convolutional layer. As it is a fully convolutional architecture, the output resolution of each layer varies, depending on the resolution of the input frames.

action recognition network.

While we mainly focus on improving optical flow with the auxiliary task of next frame prediction, we also show benefits on next frame prediction.

2. Related Work

Since the work by Horn & Schunk [9], optical flow estimation has been dominated by variational methods [2, 20, 22].

The FlowNet by Dosovitskiy *et al.* [6] was the first deep network trained end-to-end on optical flow estimation. It was followed by Teney *et al.* [25] and Tran *et al.* [26]. These supervised learning methods require training data with optical flow annotations. In Dosovitskiy *et al.* [6] and Mayer *et al.* [17] synthetic datasets were introduced to provide such data. Tran *et al.* [26] applied an existing variational method to create pseudo-ground truth data.

Instead, Ahmadi and Patras [1] and Yu *et al.* [32] formulated the task as an unsupervised learning problem. To this end, they used a cost function based on the classical color constancy assumption, as it is used in variational techniques.

Video prediction has been very popular recently [16, 31, 7, 14, 10, 15, 23, 19]. Although some of these works focus on prediction as the main objective [16, 31], most of them use it as an auxiliary task. Finn *et al.* [7] proposed an action-conditioned video prediction model to facilitate unsupervised learning for physical interaction. Patraucean *et al.* [19] learn optical flow by warping the current frame to the next one. Lotter *et al.* [14] use prediction to learn representations for object recognition.

The works by Pinteau *et al.* [21], Walker *et al.* [28, 29], Jayaraman *et al.* [10], and Vondrick *et al.* [27] focus on motion prediction. Their predicted motion is conditioned on a single input frame. In contrast, we model future motion

based on current motion and the scene content by making explicit use of two consecutive frames as input.

3. Hybrid Architecture and Training Schedule

3.1. Optical Flow Estimation

The flow estimation network in the proposed hybrid architecture largely builds on the FlowNet architecture introduced in Dosovitskiy *et al.* [6]. As illustrated in Section 5, we add two more up-convolutional layers (Upconv1, Upconv0) to the decoder. This yields a flow field with the resolution of the input images. This is advantageous when combining the network with next-frame prediction. In contrast, the network in Dosovitskiy *et al.* [6] yields a lower resolution flow field, which is up-sampled with bilinear interpolation.

In Section 5, the first and second row compose the encoder and decoder components of the flow estimator respectively. While our network follows the same multi-resolution scheme as in Dosovitskiy *et al.* [6], in Section 5 we omit the extra details regarding the so-called refinement steps (Figure 3 of [6]) which represent lower-resolution outputs. We use the endpoint error loss (EPE) for training of this branch of the network. A more detailed illustration of the architecture is provided in the supplementary material.

3.2. Next-Frame Prediction

The network for the auxiliary task of next-frame prediction shares the encoder with the flow estimation network, and adds a second decoder stream with independent weights but using the same architecture. Rows 1 & 3 in Section 5 form the next-frame prediction component of the network. As suggested in previous work [16], we use an L1 loss to avoid blur in the generated images.

As reported in section 4, we experimented with different number of frames as input for next-frame prediction. How-

ever, in the multi-tasking scheme, we only use a 2-frame set-up to be compatible with the paired task of flow estimation. The two three-channel RGB images are provided as a stacked six-channel input to the overall network.

3.3. Joint training

For joint training of the hybrid network, there are two challenges. First, the data comes from two different sources, and there are multiple ways how to mix them during training. Secondly, unlike synthetic data, the real data does not come with optical flow ground truth, i.e., for real data as input, there is no loss for the flow related stream of the network.

Hybrid data We mix the data at the minibatch level: data in a single batch is taken completely either from the synthetic dataset or the real-world dataset. The minibatch $\mathcal{B}^{(i)}$ at iteration i alternates between minibatches $\mathcal{B}_1^{(i)}$ & $\mathcal{B}_2^{(i)}$ from the two data sources

$$\mathcal{B}^{(i)} = (1 - s^{(i)})\mathcal{B}_1^{(i)} + s^{(i)}\mathcal{B}_2^{(i)} \quad (1)$$

using the *switch* function

$$s^{(i)} = \lfloor \frac{i \bmod (n_1 + n_2)}{n_1} \rfloor, \quad (2)$$

which always yields 0 or 1 and allows for different numbers of cycles n_1 and n_2 dedicated to each data source, respectively.

Batch-variant loss The total loss at the i^{th} iteration is computed according to:

$$\mathcal{L}^{(i)} = w_1\mathcal{L}_1^{(i)}s^{(i)} + w_2\mathcal{L}_2^{(i)} \quad (3)$$

in which $\mathcal{L}_1^{(i)}$ & $\mathcal{L}_2^{(i)}$ are the flow and frame estimation losses respectively, with their assigned weights, w_1 & w_2 .

In case of real-data without ground truth optical flow, we deactivate the flow related loss $\mathcal{L}_1^{(i)}$ and, thus, the flow related decoder stream of the network. Both the loss and the loss gradient are set to zero. We keep the loss in sync with the switch function s to ensure the desired functionality: the network learns on both tasks when synthetic data is provided, but skips updating the optical flow decoder when there is no ground truth.

We set the loss weights such that w_1/w_2 is equal to σ_2/σ_1 where σ 's are the estimates of the variances of the input data (frame vs. flow) and are computed over a subset of 500 random samples from the training sets. In the following experiments, we report results with a fixed ratio of 1/5 for w_{flow}/w_{frame} .

In our main experiments we fix the ratio of cycles dedicated to synthetic and real data sources. But we also provide an analysis on the effect of different cycle ratios on the quality of the output flow field.

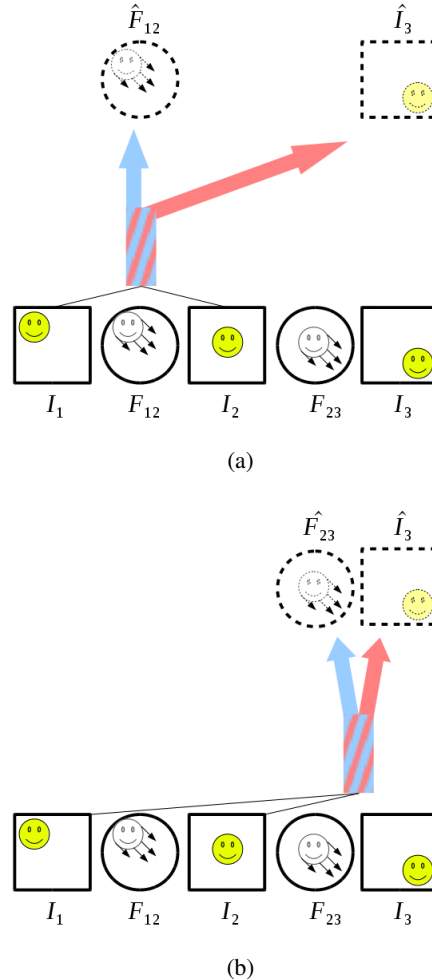


Figure 3: Illustration of the two multi-tasking schemes introduced in this paper. (a) Combination of flow estimation with next-frame prediction; (b) the next-flow prediction replaces the task of current flow estimation. Each I_k denotes a single frame in a video sequence of length 3, and F denotes a flow field. In both scenarios only I_1 and I_2 are the inputs to the network. Therefore, the terms “current flow” and “next-flow” refer to F_{12} and F_{23} , respectively.

3.4. Next-Flow Prediction

In a multi-tasking scheme, for the combination to yield significant improvements in the results, the two tasks need to be “related” [4]. In context of the current work, we hypothesize that prediction of the “next-flow” (i.e. the future flow to come), may have more in common with the task of next-frame prediction. Figure 3 compares the two combinations and gives an intuition on how the two “prediction” tasks match.

More formally: The two tasks share the encoder component of the network, that maps $(I_1, I_2) \mapsto z$, where z is the

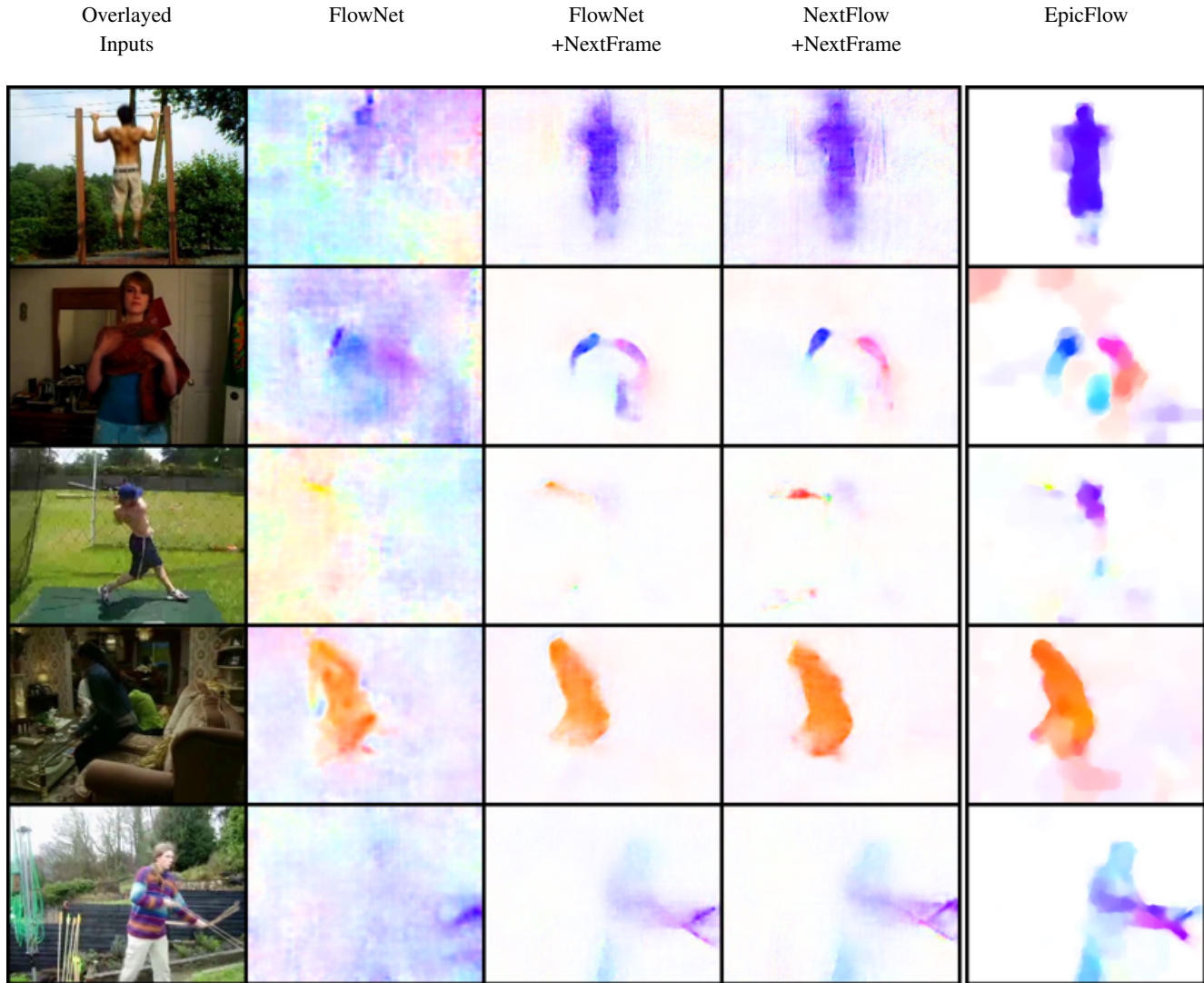


Figure 4: Some samples from the estimated optical flow fields on real scenes from HMDB51 [11]. Both of our suggested methods in the middle columns, show clear improvements in the flow fields and preserving the object shapes.

internal representation to be learned by the network. This mapping is affected by both tasks during the backward pass. For multi-tasking to make sense, we expect that some features learned by the encoder are beneficial for both target tasks. We hypothesize that the pair of $(z \mapsto I_3, z \mapsto F_{23})$ have more to share, compared to $(z \mapsto I_3, z \mapsto F_{12})$ not only due to both being “prediction” tasks, but also because in obtaining the future flow, the network needs to learn, at least implicitly, about the future frame. This is not the case for the current flow (Figure 3). This hypothesis is supported by our experimental results.

3.5. Training Details

We train the network for 1 million iterations, with a batch size of 8 for both of the data sources. The initial learning

rate is 0.0001, and drops by a factor of 0.5 every 100K iterations starting from 300K. We use ADAM [13] for optimization with $\beta_1 = 0.9$, $\beta_2 = 0.999$. On an NVIDIA Titan X, training takes roughly 10 days.

4. Experiments

4.1. Datasets

For hybrid training of the network we need 2 datasets per experiment. We used the so-called “FlyingThings3D” dataset of Mayer *et al.* [17] as the data source with ground truth optical flow. It consists of more than 20000 training images and allows training a network from scratch. Moreover, it provides an independent test set that we used for testing. The much smaller Sintel dataset [3] has 1064 sam-

	Dataset for Frame Prediction	KITTI'12		KITTI'15	Sintel	FlyingThings3D
		train	test	train	train	test
B2B Unsupervised FlowNet [32]	-	-	11.3	-	-	-
FlowNet [6]	-	8.26	-	-	4.50	-
FlowNet Baseline	-	8.79	-	15.59	4.33	1.84
FlowNet Baseline+NextFrame	Sports	8.55	-	15.16	4.38	1.86
FlowNet Baseline+NextFrame	Cityscapes	8.49	-	14.68	4.24	1.80
FlowNet Baseline+NextFrame	KITTI:frames	8.37	-	14.15	4.30	1.84
FlowNet Baseline+NextFrame	Sports + KITTI:frames	8.39	-	15.08	4.29	1.86
FlowNet Baseline+NextFrame	Sports→KITTI:frames	7.78	9.2	13.95	4.36	1.85
FlowNet [6] →KITTI:flow	-	7.52	9.1	-	-	-
FlowNet+NextFrame →KITTI:flow	KITTI:frames	5.31	-	10.19	5.35	2.82

Table 1: Quantitative evaluation of optical flow estimation performance based on End Point Error (EPE). “KITTI:frames” indicates video frames (without flow annotations) from the KITTI dataset. Moreover, wherever the evaluation is performed on a KITTI (2012/2015) training subset, the data used for the training of the network is taken from its counterpart (2015/2012). The → sign indicates a pre-training/fine-tuning process.

ples and was used only for testing.

The only available real-world dataset with ground truth optical flow is the KITTI dataset. There are two independent datasets, KITTI 2012 [8] and KITTI 2015 [18]. We used both datasets for the quantitative evaluation of the optical flow. Since both datasets are independent, we always used one for training and the other for testing. Except for one experiment, we did not use the optical flow ground truth for training but only the images. We took the frames from the “multi-view” extension of the datasets, consisting of 4074 and 4200 images in the 2012 and 2015 versions, respectively.

There are many large real-world datasets without optical flow ground truth. We used mainly a subset of the Sports1M dataset [12] for the self-supervised training task. The subset includes all videos with a file size up to 5 MBytes, amounting to more than 220K videos and 220M frames. We will make the selection list available online. Also in another experiment, we simply used 50000 frames from videos of the Cityscapes dataset [5].

Moreover, we used the UCF101 [24] and HMDB51 [11] datasets for testing the optical flow indirectly in an action recognition scenario. The datasets contain more than 2M & 600K frames, respectively.

To compare the performance of our next-frame predictor to published work, we used the same subset of UCF101 [24] as Mathieu et al. [16]. It consists of 387 videos.

4.2. Direct Evaluation of the Optical Flow

In Figure 4 we visualize some of the flow fields estimated with our method, FlowNet [6], and an accurate but slow variational method (EpicFlow [22]). On real-world scenes, our flow fields capture the shape of moving objects much better than the baseline FlowNet. We believe this sharp-

ness is a result of asking for pixel-level accurate results in the auxiliary task of frame prediction, which regulates the blurring that the flow branch tends to exert.

We quantitatively evaluated the method on KITTI 2012 & 2015. Table 1 shows these results along with two synthetic datasets. All the experiments used the same synthetic source of data and they differ only in the source of real data. ‘FlowNet Baseline’ is our full-resolution extension of the architecture of [6] trained on FlyingThings3D. ‘FlowNet+NextFrame’ indicates our hybrid multi-tasking scheme.

Results from various configurations are displayed in Table 1. Although the Sports dataset has little similarity with the scenes in KITTI, using this data for the auxiliary task yields significant improvements on KITTI. Using frames from the Cityscapes dataset, improves the results even more, as the videos are recorded in a similar context to that of KITTI’s. There is no significant change on the synthetic datasets. This does not come as a surprise, since the FlyingThings3D dataset can cover other synthetic datasets like Sintel well. There is no significant domain shift from the training set to the test set in this case.

Using video frames from the KITTI dataset (labeled as ‘KITTI:frames’) rather than the Sports or Cityscapes datasets for the auxiliary task, improves results on KITTI as expected. We also experimented with combining the two real datasets, both in a parallel fashion and in a pre-training/fine-tuning scheme (‘Sports→KITTI:frames’). The latter led to another large improvement. We submitted this version to the official KITTI evaluation site to obtain results on the KITTI test set. The result is essentially as good as the FlowNet fine-tuned on KITTI. Figure 5 depicts a qualitative comparison on this benchmark.

We also report results on the fine-tuned FlowNet com-

		Action Accuracy (%)	
		UCF101 [24]	HMDB51 [11]
Classical	EpicFlow [22]	82.8	56.1
	TV-L1 [20] (as reported in [30])	87.2	-
CNN based	FlowNet	62.0	38.6
	FlowNet pre-trained with NextFrame	63.4	38.4
	FlowNet+NextFrame Multi-tasking (1:5)	74.1	48.4
	NextFlow+NextFrame Multi-tasking (1:5)	75.5	48.9

Table 2: Action classification accuracy. Each row contains results of training and testing the action classifier on optical flow generated by a specific method. 1:5 indicates the real to synthetic iterations ratio.

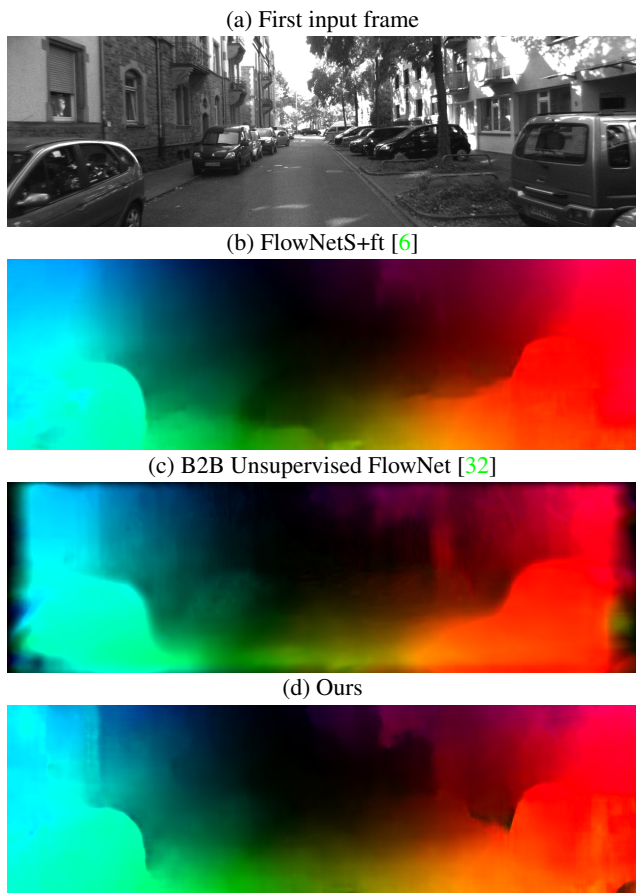


Figure 5: Sample result on the KITTI benchmark. The unsupervised method of Yu *et al.* [32] has problems near the image boundaries and reveals blurred motion boundaries. Our method shows a similar quality as FlowNetS+ft, although it has been only fine-tuned on unlabeled data.

combined with the hybrid learning on the auxiliary task at the very bottom of Table 1. This experiment shows that even when fine-tuning the FlowNet baseline on KITTI, hybrid training still yields significant improvements.

$n_{real} : n_{synth}$	EPE	Action Accuracy (%)	
	KITTI'12	HMDB51	UCF101
FlowNet	8.88	38.6	62.0
1:9	8.76	48.0	75.3
1:5	8.55	48.4	74.1
1:3	8.78	47.3	74.7
1:1	8.94	48.3	76.6
4:1	10.35	48.0	-

Table 3: Analysis of the effect of different cycles ratios on optical flow quality. For EPE, lower values are better. For action class accuracy, higher numbers are better.

4.3. Indirect Evaluation: Action Classification

As real-world videos rarely come with optical flow ground-truth (KITTI being an exception), possibilities for a direct evaluation of the optical flow is limited. Thus, we use the evaluation on flow-based action classification as an indirect quantitative measure on two larger real-world datasets. We use the action classifier network of Wang *et al.* [30] and train/test it with optical flow from different optical flow methods as input.

Table 2 shows the results of this evaluation. We used the Sports dataset to provide unsupervised data. The hybrid learning was done with a ratio of 1:5 for real to synthetic cycles. The optical flow with our hybrid learning scheme improved results on action recognition by a large margin (12.1% on UCF and 9.8% on HMDB) when compared to the baseline FlowNet. We achieved even larger improvements by replacing current flow with ‘NextFlow’.

We also tried a pre-training/fine-tuning scenario in which the network is initially trained for the frame prediction task (on real data), and then fine-tuned with the main task (“FlowNet pre-trained with NextFrame”). Results confirm that this sequential learning is not sufficient. The multi-tasking scheme is necessary to make good use of the auxiliary task on the real data.

We report also number of two variational methods, TV-L1 [33] and EpicFlow [22]. They provide a higher accu-

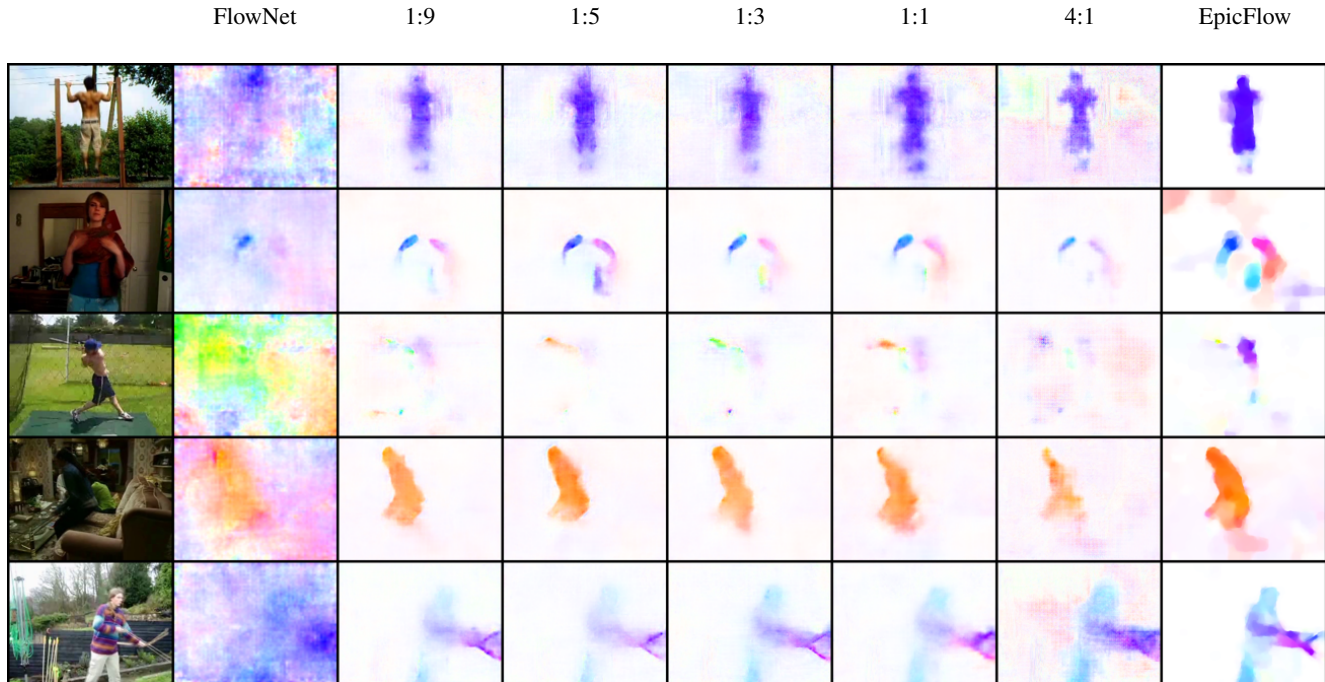


Figure 6: Qualitative comparison of different data source combination cycles. On top of each column the $n_{real} : n_{synth}$ ratio is displayed. When there is no real data involved (FlowNet), the network fails to estimate an acceptable flow field in real scenes. On the other hand, if training spends too many cycles only on frame prediction, as in the 4:1 column, the network no longer focuses enough on the optical flow task. The best results are obtained with a ratio of 1:5 or 1:3.

Method	Whole Image		Moving Regions		
	Similarity PSNR(dB)	Sharpness (dB)	Similarity PSNR(dB)	Sharpness (dB)	
Mathieu <i>et al.</i> [16]	L1	22.3	18.5	28.7	24.8
	GDL + L1	23.9	18.7	29.9	25
	Adv + GDL + L1	29.6	20.3	32	25.4
Ours (2-frame)	L1	29.9	20.6	31.9	25.4
Ours (4-frame)	L1	30.8	20.8	31.9	25.4

Table 4: Next frame prediction on UCF101 [24]. With just a simple L1 loss we already obtain clear improvements over the state-of-the-art.

racy, but are also much slower than the network based approaches.

4.4. Impact of Task Combination Cycles

We evaluated on which ratio of training cycles on synthetic and real data one obtains the best performance and on how robust the method is to deviations from the optimal ratio. We used the Sports dataset as data source in this experiment. Figure 6 shows the results for various $n_{real} : n_{synth}$ cycle ratios. Results are robust for a large range of ratios. Lower ratios approach the results of FlowNet, as the effect of the auxiliary task starts to vanish. Putting too much em-

phasis on the auxiliary task introduces artifacts in the optical flow field, since the network starts to care mostly about next frame prediction. In general, the ratio should be biased towards the supervised optical flow task. A ratio of 1:5 seems a good choice in general.

4.5. Next-Frame Prediction as a Single Task

We also evaluated the output of our next-frame prediction network and tested it on UCF. To this end, we trained it as an independent single-task network ($n_{synthetic} = 0$). Table 4 shows a comparison with Mathieu *et al.* [16] – which to the best of our knowledge is the current state-of-the-art in



Figure 7: Next frame prediction samples. Results of Mathieu *et al.* [16] are often a bit sharper due to the adversarial loss, yet the method also introduces distortions and artifacts; see the last two samples. Our next frame predictions are blurrier due to relying only on the L1 loss, but yield robust predictions without distortion. This explains the on-par quantitative results in Table 4.

next-frame prediction on UCF. Without the use of any auxiliary cost functions, as introduced in Mathieu *et al.* [16] for the sake of sharp results, and just with a single L1 loss, we obtain results on par with Mathieu *et al.* on the moving regions of the image, and significantly better results on the whole image. This means that the network is more successful on applying the motion only to the dynamic areas and keeping the static areas intact. We show qualitative examples of the predicted frames in Figure 7 and in a video in the supplementary material.

Since frame prediction has only been an auxiliary task in the network, the input settings (particularly the cycles ratio) have been set to focus on improvement of the optical flow output. Therefore, by increasing the number of flow cycles, the next-frame prediction accuracy is degraded.

5. Conclusions

We have presented a way to improve a deep network for optical flow estimation on real data by training it with an additional self-supervised auxiliary task. Our experiments showed a consistent improvement of the optical flow qual-

ity on real-world data. Thus, we believe that this approach largely improves the transfer of deep networks trained on synthetic dataset to domains in the real world. While we focused here on optical flow, the concept may transfer also to similar problems, such as disparity estimation, and alternative self-supervised auxiliary tasks.

Acknowledgments

We acknowledge funding by the ERC Starting Grant VideoLearn.

References

- [1] A. Ahmadi and I. Patras. Unsupervised convolutional neural networks for motion estimation. *arXiv:1601.06087 [cs]*, Jan. 2016. 2
- [2] T. Brox, A. Bruhn, N. Papenberger, and J. Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *Computer Vision - ECCV 2004*, pages 25–36. Springer, Berlin, Heidelberg. 2
- [3] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In

- A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012. 4
- [4] R. Caruana. Multitask Learning. In S. Thrun and L. Pratt, editors, *Learning to Learn*, pages 95–133. Springer US, 1998. 3
- [5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. pages 3213–3223. 5
- [6] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow With Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015. 1, 2, 5, 6
- [7] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances In Neural Information Processing Systems*, pages 64–72, 2016. 2
- [8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 5
- [9] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185–203, Aug. 1981. 2
- [10] D. Jayaraman and K. Grauman. Look-Ahead Before You Leap: End-to-End Active Recognition by Forecasting the Effect of Motion. In *Computer Vision – ECCV 2016*, pages 489–505. Springer, Cham, Oct. 2016. 2
- [11] H. Zhuang, H. Garrote, E. Poggio, T. Serre, and T. Hmdb. A large video database for human motion recognition. In *Proc. of IEEE International Conference on Computer Vision*, 2011. 4, 5, 6
- [12] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale Video Classification with Convolutional Neural Networks. In *CVPR*, 2014. 5
- [13] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. 4
- [14] W. Lotter, G. Kreiman, and D. Cox. Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning. *arXiv:1605.08104 [cs, q-bio]*, May 2016. 2
- [15] R. Mahjourian, M. Wicke, and A. Angelova. Geometry-Based Next Frame Prediction from Monocular Video. *arXiv:1609.06377 [cs]*, Sept. 2016. 2
- [16] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv:1511.05440 [cs, stat]*, Nov. 2015. 2, 5, 7, 8
- [17] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 2, 4
- [18] M. Menze and A. Geiger. Object Scene Flow for Autonomous Vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 5
- [19] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. *arXiv:1511.06309 [cs]*, Nov. 2015. 2
- [20] J. S. Pérez, E. Meinhardt-Llopis, and G. Facciolo. TV-L1 optical flow estimation. *Image Processing On Line*, 2013:137–150, 2013. 2, 6
- [21] S. L. Pintea, J. C. van Gemert, and A. W. M. Smeulders. Déjà Vu. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, number 8691 in Lecture Notes in Computer Science, pages 172–187. Springer International Publishing, Sept. 2014. 2
- [22] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2015. 2, 5, 6
- [23] M. Saito and E. Matsumoto. Temporal Generative Adversarial Nets. *arXiv:1611.06624 [cs]*, Nov. 2016. 2
- [24] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv:1212.0402 [cs]*, Dec. 2012. 5, 6, 7
- [25] D. Teney and M. Hebert. Learning to Extract Motion from Videos in Convolutional Neural Networks. *arXiv:1601.07532 [cs]*, Jan. 2016. 2
- [26] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Deep End2End Voxel2Voxel Prediction. pages 17–24, 2016. 2
- [27] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating Visual Representations From Unlabeled Video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–106, 2016. 2
- [28] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An Uncertain Future: Forecasting from Static Images Using Variational Autoencoders. In *Computer Vision – ECCV 2016*, pages 835–851. Springer, Cham, Oct. 2016. 2
- [29] J. Walker, A. Gupta, and M. Hebert. Dense Optical Flow Prediction from a Static Image. *arXiv:1505.00295 [cs]*, May 2015. 2
- [30] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016. 6
- [31] T. Xue, J. Wu, K. Bouman, and B. Freeman. Visual Dynamics: Probabilistic Future Frame Synthesis via Cross Convolutional Networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 91–99. Curran Associates, Inc., 2016. 2
- [32] J. J. Yu, A. W. Harley, and K. G. Derpanis. Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness. *arXiv:1608.05842 [cs]*, Aug. 2016. 2, 5, 6
- [33] C. Zach, T. Pock, and H. Bischof. A Duality Based Approach for Realtime TV-L1 Optical Flow. In *Pattern Recognition*, pages 214–223. Springer, Berlin, Heidelberg. 6

Supplementary Material

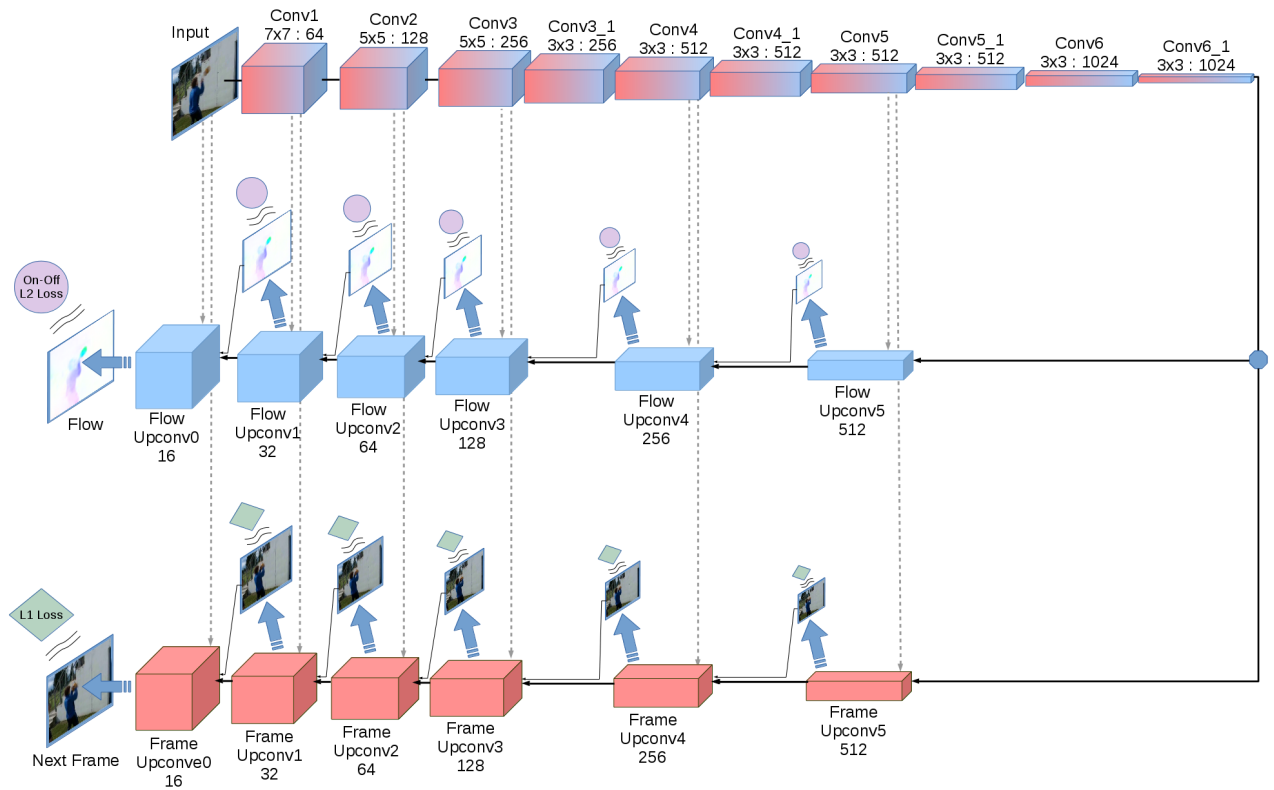


Figure 8: In this detailed illustration of the architecture, we show the contraction throughout the encoder in the first row, followed by the two decoder branches, showing the expansion in the network. Captions above/below the boxes show the layer names, as well as the number of outputs/feature maps. In the first row we also show the kernel sizes for each layer, while we do not display the fixed kernel size of 4×4 for the Upconv layers of rows 2 and 3 (see table 5). The fully-convolutional architecture can be used with different input sizes, and thus no resolution is displayed in this figure. Each \uparrow represents a convolutional layer with a kernel size of 3×3 , and stride and padding values of 1, which preserves the spatial dimensions, and maps its higher dimensional input blob to a flow or frame prediction. These low resolution predictions are then up-sampled and concatenated to the input of the next layer, along with the corresponding features from the encoder. The “On-Off” losses of the flow prediction branch are all synchronized to (de)activate the branch when necessary.

	Conv1	Conv2	Conv3	Conv3_1	Conv4	Conv4_1	Conv5	Conv5_1	Conv6	Conv6_1
kernel size	7x7	5x5	5x5	3x3	3x3	3x3	3x3	3x3	3x3	3x3
stride	2	2	2	1	2	1	2	1	2	1
padding	3	2	2	1	1	1	1	1	1	1
	Upconv5	Upconv4	Upconv3	Upconv2	Upconv1	Upconv0				
kernel size	4x4	4x4	4x4	4x4	4x4	4x4				
stride	2	2	2	2	2	2				
padding	1	1	1	1	1	1				

Table 5: Kernel size, stride and padding settings for different layers of the network. Upconv layers in the flow and frame branches share the same settings.

$n_{real} : n_{synth}$	∞	8:1	4:1	1:1	1:3	1:5	1:9
Similarity - PSNR (dB) - Whole Image	29.9	29.20	29.14	28.8	28.57	28.34	28.24

Table 6: Analysis of the effect of different cycles ratios on frame prediction. The more flow prediction cycles we include in the arrangement, the lower the prediction quality goes.