UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss

Hauptseminar: Recent Advances in Computer Vision

Soumyadeep Bhattacharjee

Abstract—The increasing predominance of end-to-end deep learning methods in the field of Computer Vision has necessitated the need for large quantities of labelled data. However, procuring dense per-pixel ground truth in the optical flow setting, for real world scenes is difficult, owing to the high setup and infrastructure costs involved, making such data infrequent. To circumvent the need for ground truth flow, this paper proposes an unsupervised loss based on occlusion-aware bidirectional flow estimation and a robust census transform. This approach outperforms previous unsupervised deep networks on the KITTI benchmarks by a substantial margin. It even performs better than existing supervised methods trained only on synthetic datasets. Further fine-tuning allows the method to achieve competitive optical flow accuracy on the KITTI 2012 and 2015 benchmarks, which enables an optional generic pre-training of supervised networks for datasets for which limited ground truth data is available.

1 Introduction

Optic Flow is the inter-frame displacement at each pixel in an image resulting from spatio temporal brightness variations, produced by the relative motion between an observer and the scene. Estimation of Optic Flow is a classical problem in Computer Vision and has applications in object detection and motion tracking. Traditional methods like Lucas-Kanade, Horn-Schunck and other variational approaches have already addressed the problem well.

However, standard energy-based optical flow methods require expensive optimization at test time and is not the ideal choice for real-time applications. Hence, deep-learning techniques in the Optic Flow setting are being increasingly explored to tackle this problem. Realistic benchmarks like MPI Sintel have continuously challenged the traditional methods, thereby a growing need for learning methods in estimating Optic Flow have emerged as a result.

Nevertheless, deep learning methods come at a cost - they rely on the availability of large amounts of data with ground truth optical flow for supervised learning to train the millions of hyperparameters involved in the learning process. As an alternative, synthetic datasets can be used, for which ground-truth data is easy to obtain in abundance. However, we may have to trade realism for quantity. This is because there exist intrinsic differences between real and synthetic imagery. The limited variability of synthetic datasets, less complex lighting patterns lead to a poor generalization to real world scenes and overcoming this remains a challenge.[1][5]

In order to address this issue, optic flow networks based on unsupervised learning have been proposed in recent work (Ahmadi and Patras 2016; Yu, Harley, and Derpanis 2016; Ren et al. 2017), where training is only performed on the original image sequences, thereby side-stepping the need for ground truth flow. Although unsupervised learning is a promising advancement towards removing the dependency on synthetic data, these approaches do not surpass the accuracy of supervised methods.[6, 9]

This paper proposes an unsupervised loss taken as a cue from classical energy-based optical flow methods. The loss is based on occlusion-aware bidirectional flow estimation and a robust census transform. The aim of this paper is to investigate possible ways for improving the accuracy of unsupervised learning for optic flow and to uncover whether unsupervised methods are a viable alternative or an addition to supervised learning. The architecture of the network is built upon recent optical flow CNNs (Dosovitskiy et al. 2015; Ilg et al. 2017). The supervised method on synthetic data is replaced by an unsupervised photometric reconstruction loss similar to (Yu, Harley, and Derpanis 2016). Bidirectional optical flow is computed both in forward and backward direction, by performing a second pass with the two input images interchanged. A loss function based on the bidirectional flow is designed to allow occlusion handling [3]. For robustness on real images, a ternary census transform is performed to allow illumination invariance. The design choices behind the unsupervised loss are validated by a comprehensive ablation study of the independent loss terms. This unsupervised model outperforms previous unsupervised deep networks by a very large margin on the KITTI benchmarks (Geiger, Lenz, and Urtasun 2012; Menze and Geiger 2015). It also surpasses architecturally similar supervised approaches trained completely on synthetic data. Following the unsupervised training on a large amount of realistic data, sparse ground truth, if available, can be used to refine the estimates in areas that are intrinsically difficult to penalize in an unsupervised manner, such as at motion boundaries.

Institut für Visualisierung und Interaktive Systeme, Universität Stuttgart. Summer Semester 2019.

2 Related Work

2.1 Supervised Learning

The first end-to-end supervised learning for optical flow was introduced with FlowNet (Dosovitskiy et al. 2015) which used an encoder-decoder network with two consecutive input images to produce a dense optical flow map.

A large augmented synthetic dataset of animated 3D chairs was used for training. The limited realism of the training dataset did not allow FlowNet to perform well on the benchmarks. A more accurate and complex family of supervised networks, FlowNet2 was introduced by Ilg et al. 2017 which improved upon the original architecture by stacking multiple FlowNet networks thereby producing iterative refinement. FlowNet2 achieves state-of-the-art accuracy by fine-tuning on the sparse ground truth from the KITTI 2012 (Geiger, Lenz, and Urtasun 2012) and KITTI 2015 (Menze and Geiger 2015) training sets.

Some other deep architectures also exist, which are based on various paradigms of supervised learning. Gadot and Wolf, 2016 proposed patch matching which uses a Siamese CNN to independently and parallelly compute the descriptors of both images, which are then compared using the L2 norm. Other methods include discrete optimization (Guney and Geiger 2016), and coarse-to-fine estimation (Ranjan and Black 2017). This paper focuses on the FlowNet architecture and its variants, but the unsupervised loss can be combined with other network architectures that directly predict the flow, e.g. (Ranjan and Black 2017).

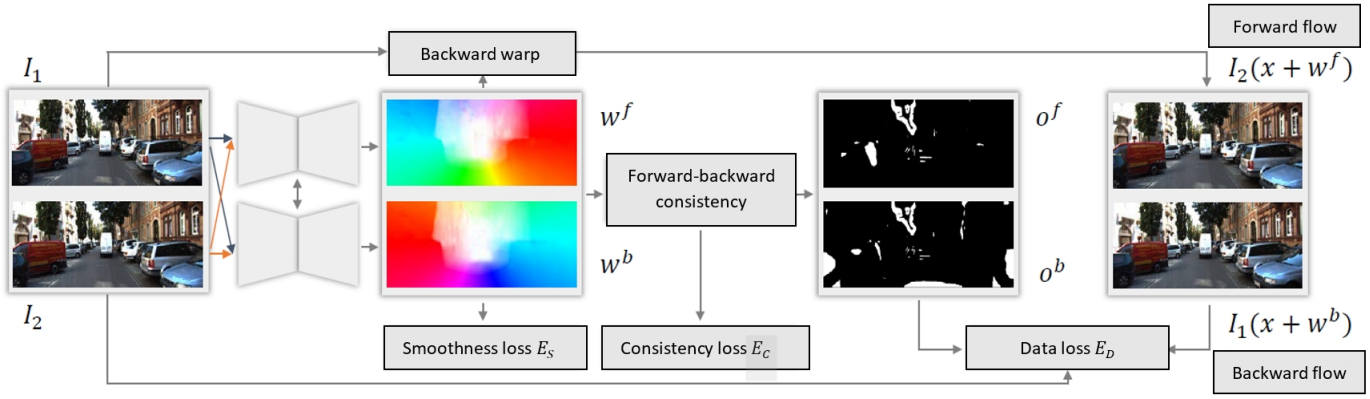

Fig. 1. The Schematic of UnFlow showing the different loss components. The data loss compares flow-warped images to the respective original images and penalizes their difference. Forward-backward consistency based on warping the flow fields, is used for estimating occlusion maps which masks the differences in the data loss. The flow fields in both directions are regularized assuming $2nd$-order smoothness. [Source: Modified from Meister et.al 2017]

2.2 Unsupervised Learning

To remove the dependency on synthetic datasets Yu, Harley, and Derpanis (2016) and Ren et al. (2017) suggested an unsupervised method based on FlowNet. The original supervised loss is replaced by a proxy loss based on the brightness constancy and smoothness assumptions which is used to train the unsupervised network on unlabelled image pairs from videos provided by the raw KITTI dataset (Geiger et al. 2013). However, this method fails to outperform the original FlowNet, probably due to the high simplicity of the proxy loss.

To overcome these issues, subsequent methods combined the unsupervised proxy loss with proxy ground truth generated by a classical optical flow algorithm (Zhu et al. 2017). Zhu and Newsam 2017, attempted to improve unsupervised CNNs by replacing the underlying FlowNet architecture. However, the improvement of all the methods over purely unsupervised approaches are minor and is still outperformed by the supervised FlowNetS.[9][6]

3 Unsupervised Learning of Optical Flow

This method uses the previous FlowNetS-based UnsupFlownet (Yu, Harley, and Derpanis 2016) and makes three major modifications:

1. A symmetric, occlusion-aware loss is designed, based on bidirectional (i.e., forward and backward) optical flow estimates.

2. The FlowNetC is trained with the comprehensive unsupervised loss to estimate bidirectional flow.

3. Iterative refinement is performed by stacking multiple FlowNet networks [4].

Additionally, an optional supervised loss can be used for fine-tuning the network on sparse ground truth data after performing the unsupervised training.

3.1 Unsupervised Loss

Given two temporally consecutive frames of an image sequence $I_1, I_2 : P \rightarrow R^3$, our goal is to estimate the optical flow $w^f = (u^f, v^f)^T$ from I_1 to I_2. The inverse optical flow $w^b = (u^b, v^b)^T$ is also needed to compute the occlusion map o^f, o^b at each pixel. A joint estimation of bidirectional flow is performed by making all loss terms symmetrical (i.e., computing them for both flow directions).

To allow for the subdifferentiable calculation of losses for use with backpropagation, bilinear sampling is employed at flow-displaced positions. This is performed by backward-warping I_2 using the flow w_f and then comparing the backward-warped second image to the first image using the bilinear sampling scheme of Jaderberg et al. (2015).

The Unsupervised loss used for training the network is the weighted sum of all the individual loss terms (Data loss, Smoothness loss and Consistency loss):

$$\begin{aligned} E(w^f, w^b, o^f, o^b) = \quad & E_D(w^f, w^b, o^f, o^b) + \quad \lambda_S E_S(w^f, w^b) \\ & + \quad \lambda_C E_C(w^f, w^b, o^f, o^b) \end{aligned} \tag{1}$$

3.1.1 Data Loss and Occlusion

The idea behind the design of the unsupervised loss is due to the fact that a non-occluded pixel in the first frame should be comparable to the pixel in the second frame to which it is mapped by the flow (Fleet and Weiss 2006). This observation does not hold for pixels that become occluded, however, as the corresponding pixels in the second frame are not visible. To avoid learning incorrect distortions, occluded pixels are masked from the data loss (Xiao et al. 2006). The occlusion detection is based on the forward-backward consistency assumption (Sundaram, Brox, and Keutzer 2010), such that pixels are marked as occluded whenever the mismatch between the forward and backward flows become too large. For non-occluded pixels, the forward flow should ideally be the inverse of the backward flow at the corresponding pixel in the second frame. The occlusion flag in the forward direction o_x^f is defined as 1 whenever the constraint

$$|w^f(x) + w^b(x + w^f(x))|^2 < \alpha_1(|w^f(x)|^2 + |w^b(x + w^f(x))|^2) + \alpha_2 \tag{2}$$

is violated, and 0 otherwise. The flag for the backward direction o_x^b is defined in the same way with w^f and w^b exchanged. The variables $\alpha_1 = 0.01$, $\alpha_2 = 0.5$ are defined for all the experiments.

The occlusion-aware data loss is defined as

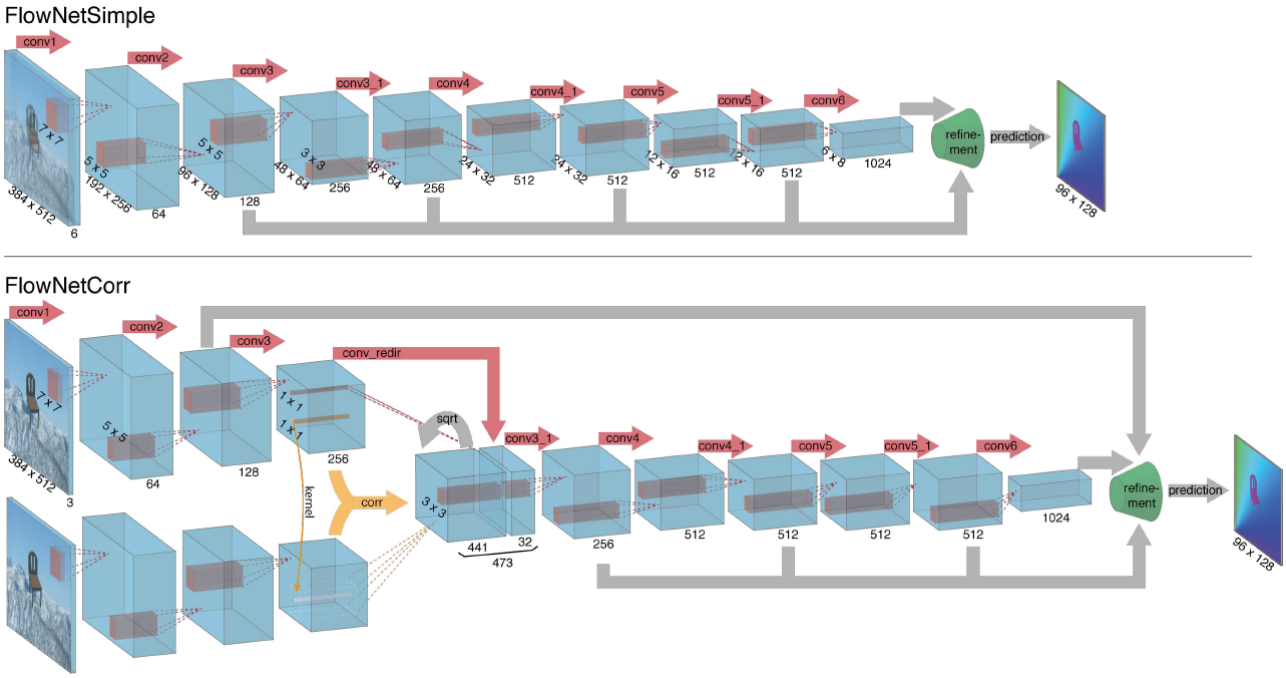

Fig. 2. The two network architectures: FlowNetSimple (top) and FlowNetCorr (bottom). The green funnel is a placeholder for the expanding refinement part shown in Fig 3. The networks including the refinement part are trained end-to-end [Source: Dosovitskiy et al. 2015]

$$E_D(w^f,w^b,o^f,o^b) = \sum_{x\in P}(1-o_x^f).\rho(f_D(I_1(x),I_2(x+w^f(x)))) + o_x^f\lambda p + (1-o_x^b).\rho(f_D(I_2(x),I_1(x+w^b(x)))) + o_x^b\lambda p \quad (3)$$

where $f_D(I_1(x),I_2(x))$ is the photometric difference between two corresponding pixels x and x', and $\rho(x) = (x^2+\varepsilon^2)^\gamma$ is the robust generalized Charbonnier penalty function [8], with $\gamma = 0.45$. The photometric difference for non-occluded pixels is penalized with a constant penalty λp, to prevent the trivial solution where all pixels get occluded. The brightness constancy constraint $f_D(I_1(x),I_2(x)) = I_1(x) - I_2(x)$ used to measure the photometric difference in the previous work (Yu, Harley, and Derpanis 2016) is not invariant to illumination changes common in real-world imagery. To overcome this issue, a ternary census transform is used.[10] The census transform provides photometric invariance against additive, multiplicative and gamma changes by preserving the intensity order within a local neighborhood.[7]

3.1.2 Smoothness Loss

To encourage collinearity of neighboring flows and attain better regularization, a second order smoothness constraint (Trobin et al. 2008; Zhang et al. 2014) is used on the flow field.

$$E_S(w^f,w^b) = \sum_{x\in P}\sum_{s,r\in N(x)} \rho(w^f(s) - 2w^f(x) + w^f(r)) + \rho(w^b(s) - 2w^b(x) + w^b(r)) \quad (4)$$

where $N(x)$ consists of the horizontal, vertical, and both diagonal neighborhoods around x and $\rho(.)$ computes the average over the original generalized Charbonnier penalties of each component.

3.1.3 Consistency loss

For non-occluded pixels, a forward-backward consistency penalty is added.

$$E_C(w^f,w^b,o^f,o^b) = \sum_{x\in P}(1-o_x^f).\rho(w^f(x) + w^b(x+w^f(x))) + (1-o_x^b).\rho(w^b(x) + w^f(x+w^b(x))) \quad (5)$$

3.2 Network Architecture

Given a dataset consisting of input image pairs, the UnFlow-C network is trained to predict the x-y flow fields directly from the images. This network is built upon FlowNetC (Dosovitskiy et al. 2015), which processes two consecutive images in two separate input streams and explicitly correlates them at a subsequent layer.

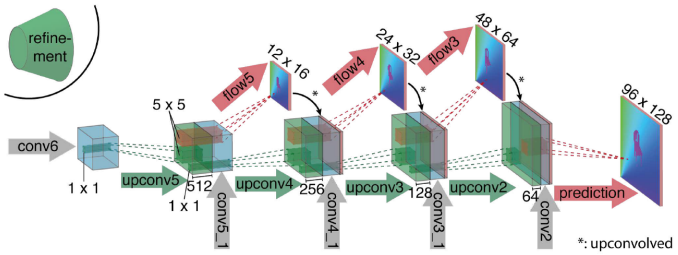

Fig. 3. Refinement Network [Source: Dosovitskiy et al. 2015]

In order to make training computationally feasible and to aggregate information over large areas of the input images, pooling is necessary in Convolutional Neural Networks. However, pooling results in reduced resolution, so in order to provide dense per-pixel predictions, the coarse pooled representation needs to be refined.[2] Hence an expanding part is used in the network which cleverly refines the flow to high resolution by concatenating it with corresponding feature maps from the 'contractive' part of the network and if available, an upsampled

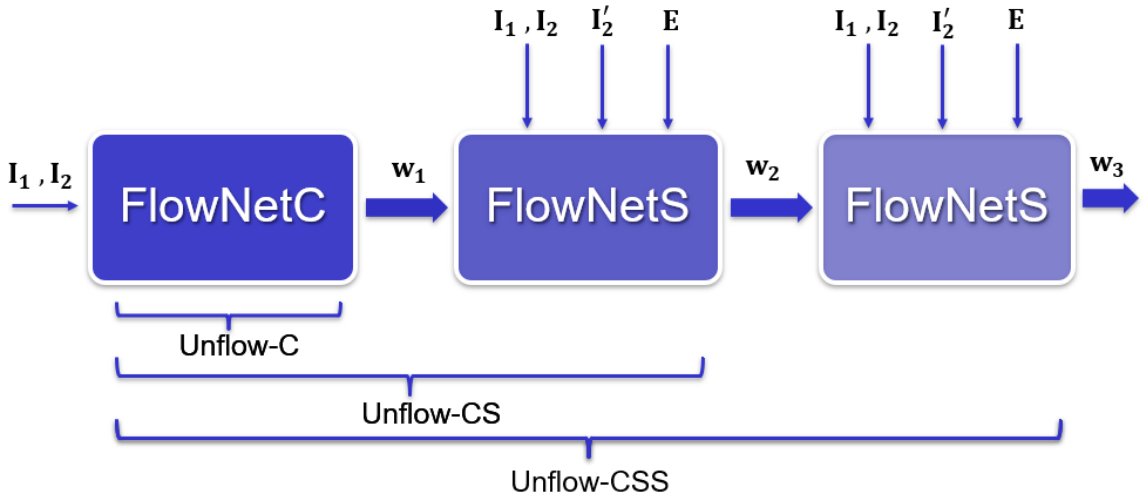

Fig. 4. Iterative refinement using multiple stacked FlowNets.The estimate of UnFlow-C is passed into a separate FlowNetS with independent weights and the two-network stack is termed as UnFlow-CS. An additional FlowNetS is concatenated after UnFlow-CS to refine its estimate and the three-network stack is termed UnFlow-CSS.

coarser flow prediction. This allows preservation of both the high-level information transferred from coarser feature maps and fine local information provided in the lower layers of the feature maps. Networks consisting of contracting and expanding parts are then trained as a single unit using backpropagation.

Similar to the supervised FlowNet, the losses are computed for all intermediate predictions from the refinement network to guide the learning process at multiple resolutions and combine them by a weighted average. The total loss is given by:

$$E_{\text{unsup}} = \sum_i \lambda_l^i E_i \tag{6}$$

To compute bidirectional optical flow, FlowNetC is applied to the RGB images (I_1, I_2) to obtain the forward flow (u_f, v_f) and the same computation is applied to (I_2, I_1) to obtain the backward flow (u_b, v_b). The weights are shared in both directions to train a universal network for optical flow.

To compute large displacement optical flow, multiple FlowNets are combined to perform Iterative refinement. Inputs are concatenated with the output from previous layers. The estimate of UnFlow-C is passed into a separate FlowNetS with independent weights and the two-network stack is termed as UnFlow-CS. Another pass for each flow direction is performed and all weights are shared between the two passes. In addition to the original images, the initial flow estimate, the backward-warped second image,the brightness error between the warped image and the first image is input into the iterated network. In the same way, additional FlowNetS is concatenated after UnFlow-CS to refine its estimate and the three-network stack is termed UnFlow-CSS.

Fine-tuning: If the ground truth is available for certain training sets, supervised finetuning can be performed wherein the network loss is computed by comparing the bilinearly upsampled final flow estimate to the ground truth flow at all pixels for which ground truth exists. This enables generic pre-training of supervised networks for datasets with limited amounts of ground truth.

$$E_{\text{sup}}(w^f) = \sum_{x \in P} v_x^f \rho(w^f(x) - w_{\text{gt}}^f(x)) \tag{7}$$

where $v_x^f = 1$ if there is valid ground truth at pixel x and $v_x^f = 0$ otherwise. The loss for only the final prediction is computed as one would want to avoid further increasing the sparsity of the ground truth, and hence prevent downsampling the ground truth flow. During fine-tuning, only the first pass of the network for the forward flow is calculated, as ground truth exists only for this direction.

4 Experiments

4.1 Training

The networks are first trained on SYNTHIA and then on KITTI raw or Cityscapes in an unsupervised manner as proposed in the paper. An optional supervised fine-tuning is then added to the best stacked networks trained on KITTI, for which ground truth from the KITTI 2012 and 2015 training sets are used. As optimizer, Adam (Kingma and Ba 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is used.

Unsupervised SYNTHIA pre-training - A mini-batch size of 4 image pairs from the SYNTHIA data is trained for 300K iterations with an initial learning rate of 1.0e-4 fixed for the first 100K iterations, then progressively dividing it by two after every 100K iterations. While training the stacked networks, each network is pre-trained while keeping the weights of previous networks fixed.

Unsupervised KITTI training- A mini-batch size of 4 image pairs from the raw KITTI data is trained for 500K iterations with an initial learning rate of 1.0e-5 fixed for the first 100K iterations, then progressively halving it after every 100K iterations.Stacking is performed in a similar way as on SYNTHIA.

Unsupervised Cityscapes training- The procedure is the same as for KITTI, except that only a single network is trained without stacking.

Supervised KITTI fine-tuning (-ft) - Fine-tuning is performed end-to-end on a mini-batch size of 4 image pairs from the KITTI training sets and an initial learning rate of 0.5e-5, which is reduced to 0.25e-5 and 0.1e-5 after 45K and 65K iterations, respectively. For validation, 20% of the shuffled training pairs are set aside and fine-tuned until the validation error increases.

Data Augmentations and Pre-Processing- The list of image pairs from KITTI are first shuffled and are cropped to 1152 x 320, SYNTHIA images to 768 x 512, and Cityscapes images to 1024 x 512. Random additive Gaussian noise $(0 < \sigma \leq 0.04)$, random additive brightness changes, random multiplicative color changes $(0.9 \leq \textit{multiplier} \leq 1.1)$, as well as contrast (from $[-0.3, 0.3]$) and gamma changes (from $[0.7, 1.5]$) to both frames are independently added as per

Fig. 5. Visual comparison of UnFlow-C on the KITTI 2015 training set. A baseline is used to compare the unsupervised loss akin to (Yu, Harley, and Derpanis 2016) (middle) to our best unsupervised loss (right). The ground truth and estimated flow (upper row), as well as input image overlay and flow error (lower row) is shown. The KITTI 2015 error map depicts correct estimates (≤ 3 px or $\leq 5\%$ error) in blue and wrong estimates in red tones. [Source: Meister et.al 2017]

Yu, Harley, and Derpanis (2016). For unsupervised training, the same random horizontal flipping and scaling is applied to both frames, and a relative random scaling of the second frame $(0.9 \leq factor \leq 1.1)$ is performed. The brightness changes are sampled from a Gaussian with $\sigma = 0.02$, while all other random values are uniformly sampled from the given ranges.

4.2 Evaluation

FlowNet needs both image dimensions to be divisible by 26. However, images from the KITTI flow benchmarks have a resolution of 1241x376, 1226x370, or 1242x375. Hence, bilinear upsampling of these input images to 1280x384 is performed for accuracy evaluation. The resulting 1280x384 flow estimates are then bilinearly downscaled to the original size for comparison with the ground truth flow maps.

As distances of the original pixels change after resampling, the components of the estimated flow are scaled according to the image downscaling factor in the respective directions. Only zero-padding the input images up to the next valid size introduces visible artifacts at the image boundaries and significantly increases the error. For other datasets, bilinear resampling is used analogously.[5]

4.3 Comparison of Unsupervised Loss

In order to compare the Unsupervised Loss with the exisitng methods, a detailed Loss Ablation study is performed to observe the effect of each modification of the unsupervised loss terms over the baseline (Yu, Harley, and Derpanis 2016) re-implementation, as shown in Table 1.

The metrics used for comparison are the Average Endpoint Error(average Euclidean distance between ground truth and estimated flow) and Fl-all (the ratio of pixels where the flow estimate is wrong by both ≥ 3 pixels and $\geq 5\%$). The census loss significantly improves upon the original brightness constancy loss (35% improvement), the second-order smoothness loss outperforms the first-order (5% improvement, 17% outlier reduction), and occlusion masking combined with forward-backward consistency decreases the overall error even further (14% improvement). The combination of all three modificatiions decreases the AEE by a significant margin compared to previous unsupervised training approaches (to ≤ 0.5x).

Tabelle 1. Loss Ablation study on KITTI 2012

Data Loss	Smoothness	Occlusion	AEE(all)	Outliers(all)
Brightness	1st-Order		7.20	31.93%
Census	1st-Order		4.66	20.85%
Census	2nd-Order		4.40	17.22%
Census	2nd-Order	Fwd-Bwd check	**3.78%**	**16.44%**

5 Results on various datasets

The best loss setup from the previous ablation study is used for unsupervised training of the final networks. UnFlow-CS and UnFlow-CSS is then fine-tuned with the supervised KITTI schedule and these models are termed as UnFlow-CS-ft and UnFlow-CSS-ft. In addition, a FlowNetC is trained on SYNTHIA and Cityscapes.

KITTI: On KITTI 2012, the purely unsupervised UnFlow-C outperforms the supervised FlowNetC and FlowNetS in all metrics, which shows the benefits of unsupervised learning of optical flow, trained on the relevant domain for coping with real-world scenes,

Method	KITTI 2012				KITTI 2015			Middlebury	
	AEE (All)		AEE (NOC)		AEE (All)	Fl-all		AEE	
	train	*test*	*train*	*test*	*train*	*train*	*test*	*train*	*test*
DDF (Güney and Geiger 2016)	–	3.4	–	1.4	–	–	21.17%	–	–
PatchBatch (Gadot and Wolf 2016)	–	3.3	–	1.3	–	–	21.07%	–	–
FlowFieldCNN (Bailer, Varanasi, and Stricker 2017)	–	3.0	–	1.2	–	–	18.68%	–	–
ImpPB+SPCI (Schuster, Wolf, and Gadot 2017)	–	2.9	–	1.1	–	–	17.78%	–	–
SDF (Bai et al. 2016)	–	2.3	–	1.0	–	–	11.01%	–	–
FlowNetS+ft (Dosovitskiy et al. 2015)	7.5	9.1	5.3	5.0	–	–	–	0.98	–
UnsupFlownet (Yu, Harley, and Derpanis 2016)	11.3	9.9	4.3	4.6	–	–	–	–	–
DSTFlow(KITTI) (Ren et al. 2017)	10.43	12.4	3.29	4.0	16.79	36 %	39 %	–	–
FlowNet2-C (Ilg et al. 2017)	–	–	–	–	11.36	–	–	–	–
FlowNet2-CSS (Ilg et al. 2017)	3.55	–	–	–	8.94	29.77%†	–	0.44	–
FlowNet2-ft-kitti (Ilg et al. 2017)	(1.28)	1.8	–	1.0	(2.30)	(8.61%)†	10.41%	0.56	–
UnFlow-C-Cityscapes	5.08	–	2.12	–	10.78	33.89%	–	0.85	–
UnFlow-C	3.78	–	1.58	–	8.80	28.94%	–	0.88	–
UnFlow-CS	3.30	–	1.26	–	8.14	23.54%	–	0.65	–
UnFlow-CSS	3.29	–	1.26	–	8.10	23.27%	–	0.65	–
UnFlow-CS-ft	(1.32)	1.9	(0.75)	0.9	(2.25)	(9.24%)	12.55%	0.64	–
UnFlow-CSS-ft	(1.14)	1.7	(0.66)	0.9	(1.86)	(7.40%)	11.11%	0.64	0.76

Fig. 6. Accuracy comparison on KITTI, Middlebury, and Sintel optical flow benchmarks. AEE: Average Endpoint Error; Flall:Ratio of pixels where flow estimate is wrong by both ≥ 3 pixels and $\geq 5\%$. The numbers in parentheses are the results of the networks on data they were trained on, and hence are not directly comparable to other results. [Source: Meister et.al 2017]

even in the absence of ground truth flow. The fine-tuned network performs similar to the more complex FlowNet2-ft-kitti (Ilg et al. 2017) without the need for a separate small-displacement network and custom training schedules.

Middlebury: In order to demonstrate that the networks generalize well to realistic domains outside the driving setting they were trained on, the performance was tested on Middlebury, where UnFlow-C and UnFlow-CCityscapes outperformed the supervised FlowNetS, and the stacked and fine-tuned variants perform between FlowNetS and the complex FlowNet2 models.

Sintel: The unsupervised networks cannot compete with FlowNetS+ft and FlowNet2 on the Sintel dataset, which in contrast are trained on various datasets in a supervised manner. However, it outperforms DSTFlow (Ren et al. 2017), which is also trained on a similar dataset in an unsupervised manner. This shows that the method not only strongly outperforms previous unsupervised deep networks for in-domain data, but also provides benefits for data from a domain on which it has not been trained on.

6 Limitations

Unsupervised learning techniques like the method presented in this paper are limited by the loss function, unlike supervised approaches, which are often limited by the amount of available ground truth data. The amount of information that can be gained from the available data is limited by how consistently the problem is modeled by the loss. This is a challenge which needs to be addressed.[5]

Another limitation of this method is that a parameter search has to be performed for the weighting between the loss terms to achieve the best results. This further increases the total computation time for training a model on a new domain for the first time. However this limitation is also shared by previous works employing an unsupervised proxy loss (Ahmadi and Patras 2016; Yu, Harley, and Derpanis 2016; Ren et al. 2017).

7 Conclusion

The method proposed in this paper provides an end-to-end unsupervised learning approach to enable effective training of CNN's on datasets for which no ground truth optical flow is obtainable, leveraging mechanisms from energy-based flow approaches such as a data loss based on the ternary census transform and second-order smoothness assumptions. This method also handles occlusion of pixels by use of bidirectional flow estimation. The Loss Ablation study show that using an accurate unsupervised loss, as proposed, is vital in order to exploit unannotated datasets for optical flow, more than halving the error on the challenging KITTI benchmark as compared to previous unsupervised deep learning approaches.

This method also makes CNNs for optical flow relevant for use in a larger variety of domains. The results demonstrate that using a large, real-world dataset together with the unsupervised loss can even outperform supervised training on challenging realistic benchmarks where only hand-picked synthetic datasets are obtainable for supervision. The unsupervised loss also provides a basis for pre-training of the network when only limited amounts of ground truth data are available. Compared to standard energy-based optical flow methods,the proposed unsupervised network avoids expensive optimization at test time. Furthermore, stochastic minimization of the loss over an entire dataset, as performed here, can avoid some of the drawbacks of optimizing a complex energy on individual inputs, which are common in classical approaches.

The results achieved by this method corroborates the fact that further research on more accurately modelled losses for unsupervised deep learning may be a provide a state-of-the-art in optical flow estimation.

Literatur

[1] A. Ahmadi and I. Patras. Unsupervised convolutional neural networks for motion estimation. *CoRR*, abs/1601.06087, 2016.

[2] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. *CoRR*, abs/1504.06852, 2015.

[3] J. Hur and S. Roth. Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. *CoRR*, abs/1708.05355, 2017.

[4] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *CoRR*, abs/1612.01925, 2016.

[5] S. Meister, J. Hur, and S. Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. *CoRR*, abs/1711.07837, 2017.

[6] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha. Unsupervised deep learning for optical flow estimation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 1495–1501. AAAI Press, 2017.

[7] F. Stein. Efficient computation of optical flow using the census transform. In C. E. Rasmussen, H. H. Bülthoff, B. Schölkopf, and M. A. Giese, editors, *Pattern Recognition*, pages 79–86, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[8] D. Sun, S. Roth, and M. J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision (IJCV)*, 106(2):115–137, 2014.

[9] J. J. Yu, A. W. Harley, and K. G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. *CoRR*, abs/1608.05842, 2016.

[10] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the Third European Conference on Computer Vision (Vol. II)*, ECCV '94, pages 151–158, Berlin, Heidelberg, 1994. Springer-Verlag.