An Empirical Evaluation Study on the Training of SDC Features for Dense Pixel Matching

René Schuster[1] Oliver Wasenmüller[1] Christian Unger[2] Didier Stricker[1]
[1]DFKI - German Research Center for Artificial Intelligence [2]BMW Group
firstname.lastname@{bmw,dfki}.de

Abstract

Training a deep neural network is a non-trivial task. Not only the tuning of hyperparameters, but also the gathering and selection of training data, the design of the loss function, and the construction of training schedules is important to get the most out of a model. In this study, we perform a set of experiments all related to these issues. The model for which different training strategies are investigated is the recently presented SDC descriptor network (stacked dilated convolution). It is used to describe images on pixel-level for dense matching tasks. Our work analyzes SDC in more detail, validates some best practices for training deep neural networks, and provides insights into training with multiple domain data.

1. Introduction

Nowadays, advances in computer vision are dominated by deep learning approaches. The impressive success on various topics and tasks for all kinds of applications catalyzes ever more research in this field. Though a principled way for learnable representations, classifiers, and regressors is endorsed, our current understanding of deep neural networks lacks behind. Networks are often handled as black boxes due to the stochastic and iterative nature of back-propagation, the un-interpretable interior of deep and wide architectures, and the increasing number of hyperparameters.

These facts lead to a conflict for complex, yet safety-critical applications like autonomous driving. On the one hand, most recent achievements for core components of self-driving cars, like perception or action planning, are enabled by deep learning. On the other hand, the robustness and reliability of these components remain unexplored which introduces high risk since neither the probability nor the possible maximum harm of wrong decisions is known.

As a result, we need networks that are more interpretable, more robust (however robustness can be defined), and less self-confident (*i.e*. providing a measure of certainty).

Moreover, part of the success of deep learning is driven by the availability of data. Astonishing results are often obtained only by increasing the amount of training data, using deeper architectures, and thus requiring even more data. Along with that, the computational effort for training increases likewise, introducing another limiting factor. While, in principle, there is nothing wrong with using more data, one has to keep in mind that data (labeled or unlabeled) is differently scarce for different domains and applications. Thus, a working model for one domain might not be transferable to another. Further, an advanced usage of only very few data is essential to limit the expensive efforts for annotation. As a conclusion, the available data should be used as efficient as possible to train more accurate and robust models in less time.

In this study, we will focus less on the selection of the architecture, but instead use an existing, shallow model that incorporates an understanding of the given problem into its design [27]. Rather, we will investigate effects of training procedures and data in the hope to derive some heuristics that can guide others when training deep neural networks.

Our use case is embedded in the context of environmental perception for automotive applications. In detail, the network under consideration is the recently presented SDC network [27] that was designed for image description to aid dense matching tasks, like in optical flow or stereo disparity estimation. Matching is a mid-level computer vision task that can be used to reconstruct geometry and estimate motions and therefore it builds the foundation for high-level perception and planning tasks which are required for advanced driver assistance systems and autonomous vehicles.

The rest of the paper is structured as follows. In Section 2, we describe some related work and introduce the relevant data sets for our experiments. The SDC feature description network that we use in our study is explained in Section 3 along with some deeper analysis. Our experiments are presented in Section 4. We summarize our results in Section 5.

Table 1: Characteristics of different data sets.

Data Set	Task	Number of Sequences	Frames per Sequence	Image Size [MP]	Color Space	Synthetic Real	Automotive Context
KITTI [20]	sf	200	1	0.46	RGB	R	yes
FlyingThings3D [19]	sf	2239	10	0.52	RGB	S	no
Driving [19]	sf	1	800	0.52	RGB	S	yes
Sintel [3]	mix	23	46	0.45	RGB	S	no
HD1K [16]	of	35	30	2.8	Gray	R	yes
Middlebury Flow [2]	of	8	1	0.25	RGB	both	no
Middlebury Stereo [22]	st	15	1	1.1 - 17.4	RGB	R	no
ETH3D [23]	st	16	1	0.31 / 0.46	Gray	R	no

2. Background

Related Work. The importance of training data and schedules for end-to-end optical flow estimation was investigated in [18, 28]. In [18], the usability of synthetic data for transfer learning (in the form of pre-training + fine-tuning) was investigated. The authors conducted a series of experiments about lighting, data augmentation, displacement statistics, simulation of realistic noise when generating synthetic images, hyperparameter tuning, and the importance of the order when training with multiple data sets. The model under review was FlowNet [7, 14]. Advanced training strategies for PWCNet [29] were presented in [28]. Here, the focus was to adjust the training process to improve generalization of the network for the Robust Vision Challenge[1].

The work in this paper conducts a similar empirical study with focus on training strategies for deep neural networks. Contrary to the previous work, our model of interest is a generic feature description network that is not restricted to the optical flow problem.

Matching Tasks. SDC [27] was presented as a generic feature descriptor that can be used for any dense matching task, *e.g.* stereo, optical flow, or scene flow matching. Finding image correspondences for these problems is related to different image pairs. For optical flow (*of*), images are matched in the temporal domain, taken with the same camera. For stereo matching (*st*), we have two distinct rectified cameras that capture images simultaneously. A combination of both (*mix*) is possible if a data set provides ground truth for optical flow and stereo disparity. If the annotations further provide a measure for the change of depth, image correspondences between stereo cameras over time (*cross*, (*cr*)) can be established. A data set that contains labels for *st*, *of*, and *cr* is capable of training full scene flow (*sf*) matching. In Section 4, we will show that these matching tasks have quite different characteristics.

Data Sets. As mentioned in the introduction, data is of utmost importance for training. Increasing effort is spent on capturing, labeling, or generation of large data sets for different domains to enable training of deeper and larger models. Generalization to unseen samples – and even more to unseen domains – remains a challenging problem for neural networks. Yet, a tendency to overcome this issue by extensive use of more and diverse data is evident in recent publications [1, 29].

For many applications it is hard, tedious, or impossible to collect labeled training data (*e.g.* optical flow) even considering manual annotation. Therefore, synthetic data sets are often used for training followed by fine-tuning on the target domain to transfer what was learned. Advantages of synthetic data generation include large scale and dense, exact ground truth annotations. However, image appearance (even if photo-realistic) might not fit the realistic data, thus increasing the problems of generalization, overfitting, and domain adaption.

One synthetic data set, that is relevant for our work, is FlyingThings3D (FT3D) [19] since it is, besideds KITTI [8, 20], the only other data set providing full scene flow labels. Especially the *Driving* subset of FT3D is relevant, because it simulates a traffic scenario. MPI Sintel [3] is also quite large and provides optical flow and stereo labels, making it a possible candidate for deep training.

Among realistic data sets, KITTI [20] is the natural choice since it provides scene flow ground truth (though sparse) in an automotive context. The data of HD1K [16] is also captured from a stereo camera mounted on a driving vehicle, but it provides only annotations for optical flow correspondences. The original SDC network was additionally trained on the other data sets that are part of the Robust Vision Challenge[1] for stereo and optical flow (Middlebury (MB) [22, 2] and ETH3D [23]). However, the latter three are not suitable for training because they are very limited in size. An overview of all these data sets is given in Table 1.

[1] www.robustvision.net

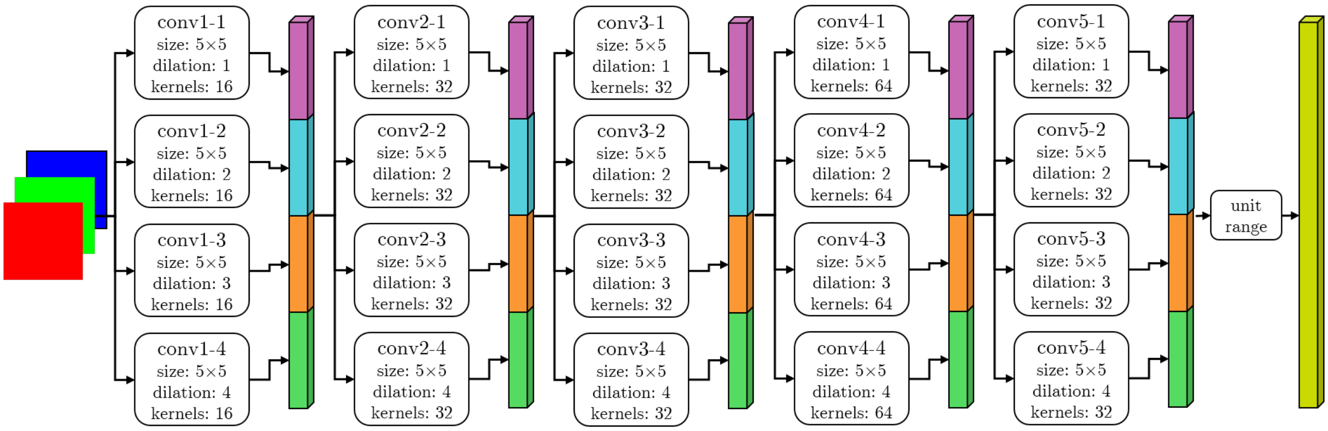


Figure 1: The architecture of the SDC feature descriptor network [27].

3. SDC Features

Network Architecture. The SDC network for feature description [27] was recently published and demonstrated superior performance over heuristic descriptors like SIFT [17] when applied in state-of-the-art matching algorithms (ELAS [9], SGM [12], CPM [13], FlowFields++ [25], and SceneFlowFields [26]). SDC was further shown to be more accurate and robust in patch matching compared to other feature networks. Its properties and the presented experiments make SDC a good candidate for generic feature computation in all kinds of architectures and applications.

The SDC network uses the concept of stacked dilated convolutions which is motivated by the observation, that dilated convolution is equivalent to regular convolution on sub-sampled input data. Therefore, concatenating the output of parallel dilated convolutions is producing a multi-scale feature representation of the input.

The proposed architecture of [27] consists of five such stacked dilated convolution layers, each with 4 parallel convolutions with 5×5 kernels and dilation rates $d = 1, 2, 3, 4$. This setup yields a receptive field of 81 pixels with a dense feature prediction for every input pixel. The complete structure of the SDC network is visualized in Figure 1.

Training. The original SDC model was trained with a mixture of data from KITTI [20], Sintel [3], HD1K [16], Middlebury (MB) [22, 2], and ETH3D [23]. The ratio of used training patches was 0.5, 0.175, 0.175, 0.05/0.025, and 0.075 respectively, which is in accordance to the variance and scale of the labeled data of each data set as shown in Table 1. The optimizer in [27] is ADAM [15] with a progressive learning rate decay (cf. Figure 6). In [27], a triplet training strategy was applied, where a reference image patch along with the corresponding and a non-corresponding patch are sampled randomly. SDC was trained with batches of 32 triplets for 1 million iterations with a thresholded hinge embedding loss [1].

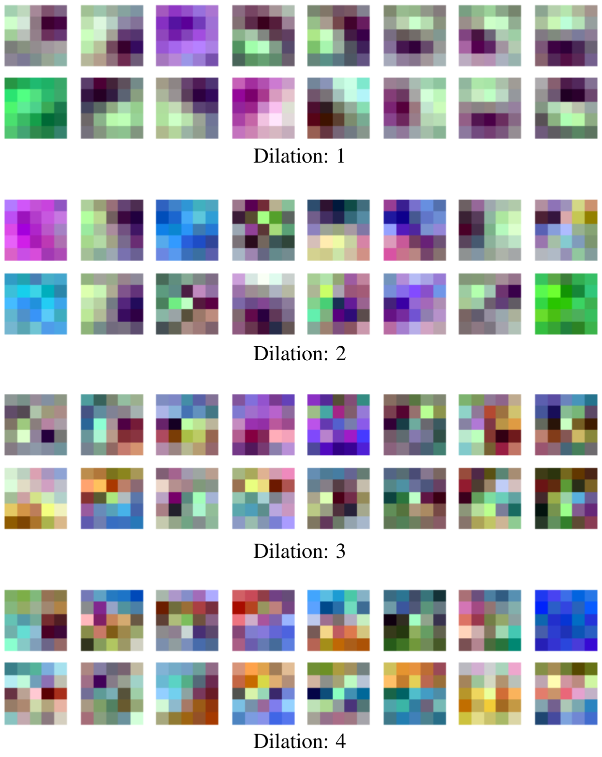
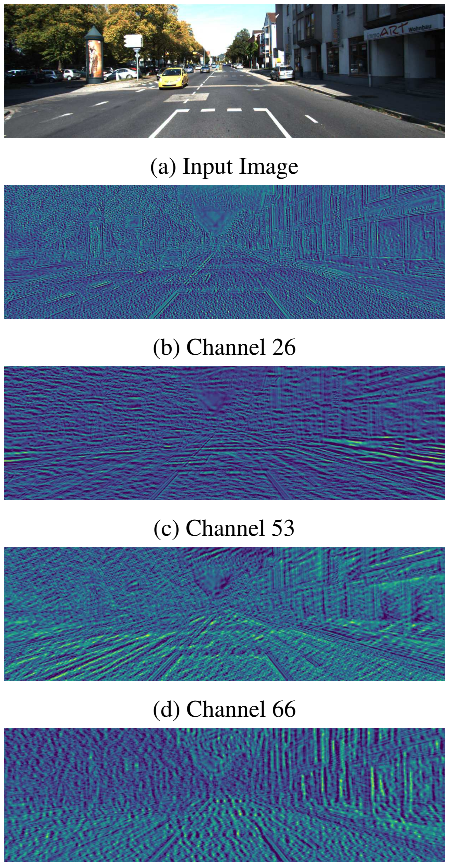


Figure 2: Convolution kernels for the first SDC layer of the SDC feature network [27]. The color gives the respective sensitivity to the RGB color channels of the input images.

Feature Analysis. The original training strategy is used for our deeper analysis of SDC features. First, we visualize the learned kernels of the first SDC layer (see Figure 2). The



(a) Input Image

(b) Channel 26

(c) Channel 53

(d) Channel 66

(e) Channel 103

Figure 3: Some SDC feature channels for the given input image.

first learned filters with a dilation rate of 1 show a high similarity to two-dimensional second order Gaussian kernels. For higher dilation rates, the kernels become less intuitive. There are also some filters that respond to a certain color.

Next, we present some of the normalized filter responses of the last SDC layer, *i.e*. the final feature representation, for an exemplary image in Figure 3. Different channels for coarse and fine structures can be identified clearly. One special observation is, that one feature channel dominates the representation, *i.e*. all values are 1, the maximum. Further experiments showed that this dimension is the same for all investigated images on all data sets. Also interesting is the fact that more than one third of all dimensions does not contribute to the description significantly, *i.e*. the features for these channels are all very close to zero for all data sets. The amount of "dead channels" decreases for increasing dilation rates (conv5-1: 18, conv5-2: 16, conv5-3: 12, conv5-4: 7). However, the remaining channels (not 0 and not 1) are all equally important for description according to their variance.

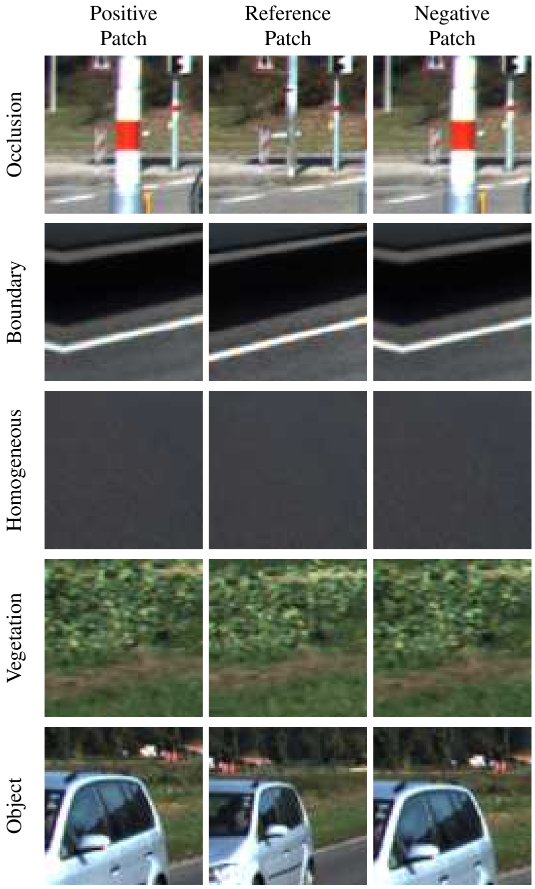
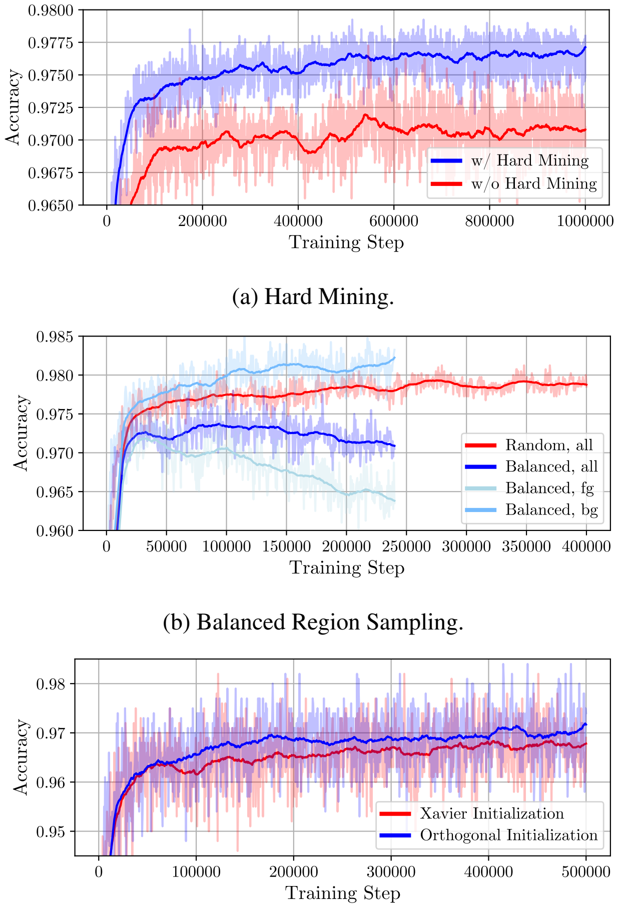


Figure 4: Misclassified triplets from the KITTI test split for the original SDC network.

Failure Cases. We evaluate SDC [27] on a test set of patch triplets. A triplet is considered as misclassified, if the feature distance of the corresponding image patches is smaller than the feature distance of non-corresponding patches (cf. Section 4). Some representative, misclassified triplets from the KITTI test set are depicted in Figure 4. The failure cases can be clustered into the following categories where one triplet can belong to multiple classes: *Vegetation* (34 %), *dynamic objects* (29 %), *occlusions* (18 %), *boundary regions* (16 %), *homogeneous patches* (14 %). While homogeneous, untextured and occluded regions can only be matched with a wider receptive field (*i.e*. changing the architecture to consider more context knowledge), the issues of dynamic foreground objects and vegetation can be tackled by changing the training schedule as done in the next sections. The only reliable way to handle image boundaries is to ignore them during feature computation and matching.

4. Empirical Study

The experiments within this section are split into two groups. First, we investigate how training can be improved



(a) Hard Mining.

(b) Balanced Region Sampling.

(c) Weight Initialization.

Figure 5: Comparison of different training setups.

in general. The second part focuses more on data and topics related to training on multiple domains. Unless stated otherwise in our experiments, a single data set model is always trained on all available image pairs (*e.g*. KITTI uses all three image pairs of the scene flow). As major evaluation criterion, the triplet accuracy is used. That is the percentage of properly distinguished patch triplets (corresponding feature distance is smaller than non-corresponding feature distance).

4.1. Improved SDC Training

Hard Mining. Hard mining is a well documented technique to speed up training and increase the accuracy especially for difficult samples [24]. It is also helpful when training with imbalanced data [6]. The idea is to ignore samples with a sufficiently accurate prediction during training and focus more on samples with less accurate or wrong predictions. In our case, we implement offline hard mining by ignoring triplets with zero loss, *i.e*. positive distance is below a threshold and the negative distance is higher than the margin (cf. the thresholded hinge embedding loss in [27]). The expected behavior of training with hard mining is threefold. First of all, we expect higher (average) losses since zero losses are neglected. Secondly, training should be speeded up because higher losses lead to higher gradients in more relevant directions. Lastly, the predictions for difficult samples should be more accurate. Figure 5a shows the validation accuracy during training with and without hard mining. Not only is the training much faster, it also reaches a higher final accuracy.

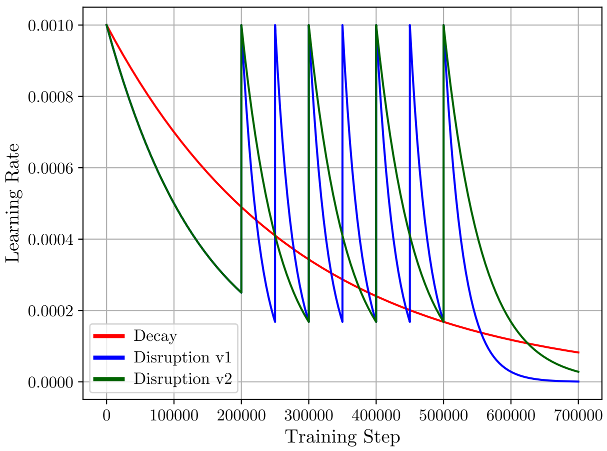


Figure 6: Monotonic decreasing learning rate schedule and two versions for learning rate disruption.

Region Sampling. Foreground objects on KITTI are one of the identified failure categories. In [27], the authors argue that this is due to the under-representation of dynamic foreground in the KITTI data set (only about 15 % of the available ground truth). Apart from hard mining, we can tackle this issue by manually balancing different image regions during patch sampling. Since ground truth object segmentation is available for KITTI training images, we can sample our reference patches for training equally often from foreground objects and static background regions. A comparison between balanced sampling and uniform random sampling is presented in Figure 5b by plotting the validation accuracy during training on KITTI optical flow data for different image regions (foreground (*fg*) / background (*bg*) / *all*). It is evident in this diagram that balanced sampling leads to very early over-fitting in foreground regions, thus hindering convergence of the model. As a result, not even the foreground regions are similarly well described as with uniform random sampling.

Initialization. The high-dimensional, highly non-linear and non-convex functional together with a stochastic iter-

Table 2: Cross evaluation for different domains represented by different data sets. For each evaluation set, the best model trained with a different data set is given in bold.

Eval	KITTI					FT3D					Driving					Sintel			HD1K	MB		ETH3D
Train	sf	mix	cr	fl	st	sf	mix	cr	fl	st	sf	mix	cr	fl	st	mix	fl	st	fl	fl	st	st
KITTI [20]	97.2	97.7	97.8	97.9	96.2	91.4	91.5	90.1	93.5	90.1	**68.6**	**70.7**	**64.5**	66.7	**75.7**	90.0	89.3	90.7	**98.5**	98.8	90.3	95.5
FT3D [19]	73.9	76.5	74.6	76.9	73.3	95.1	95.5	93.7	96.7	94.4	57.4	60.8	51.7	51.8	71.5	**95.3**	**92.7**	**95.3**	97.5	96.8	82.2	**96.7**
Driving [19]	89.3	90.8	86.7	89.6	90.6	89.9	90.0	89.1	92.0	88.5	75.2	75.8	74.2	74.6	76.7	88.7	89.0	88.7	97.0	99.2	85.9	92.7
Sintel [3]	**93.5**	**94.6**	**92.3**	**95.2**	**92.9**	**92.7**	**92.8**	**91.6**	**94.2**	**91.8**	59.8	62.7	55.2	55.6	70.8	92.3	92.7	93.0	97.2	**99.5**	**91.4**	94.2
HD1K [16]	91.0	92.0	90.6	92.8	90.8	91.6	91.7	90.7	93.7	90.3	66.2	68.2	64.0	**67.0**	69.7	88.7	88.0	88.0	99.5	99.0	89.3	95.9

ative optimization technique makes neural networks sensitive to initialization. Depending on the activation function, [11, 10] propose random initialization that considers the scale of the previous layer. Orthogonal initialization [21] was proposed for the use in linear fully-connected layers. The authors could also demonstrate positive effects with networks that use non-linear activation and convolutional layers. We compare a state-of-the-art variance scaling initializer [10] with orthogonal initialization [21] for the SDC Network in Figure 5c. Even for the shallow, fully-convolutional SDC Network with ELU activation [5], orthogonal initialization speeds up the training by about a factor of 2. The final accuracy is also slightly higher.

Learning Rate Disruption. Initialization is important for stochastic processes and so is the learning rate for the optimizer. Progressively (either in steps or continuous) decreasing learning rates are a best practice to enable convergence to local optima. However, with monotonically decreasing learning rates, the optimizer can not escape local optima. A measure to encounter this is learning rate disruption, as used *e.g.* in [28]. The idea is to disrupt the learning rate schedule by increasing the learning rate significantly (*e.g.* to the initial value) and then continue with the progressive learning rate decay. This way, the optimizer can escape from a local optimum (though not necessarily in favor of a better optimum). We have experimented with this concept when training the SDC network. Figure 6 shows three alternate learning rate schedules. The original monotonic decrease used in [27] and two variants of learning rate disruption with different periods for recovery. We could observe some signs of overfitting right after the disruption. However, the network did recover quickly but without any significant sign of changing the local optima (neither in a positive nor negative way).

4.2. Multi Domain Training

Domain Similarity. As approximation of the similarity of domains, we train mono-domain networks on a single data set and cross-evaluate them on all data sets. Table 2 shows the evaluation matrix for all trained models on all data sets. We do not train models on the Middlebury (MB) data sets [22, 2] or ETH3D [23] because of their small size. For the other data sets, we train with the union of all available image correspondences (the three scene flow image pairs for KITTI, FT3D, and Driving, and optical flow and stereo correspondences for Sintel). Training all combinations of data sets and tasks would be infeasible.

We observe that domain transfer is not necessarily symmetric. More over, the matching task (*i.e.* the type of image correspondences) has influence on the matching performance. Matching on the Driving data set is particularly difficult. On HD1K [16], matching is extremely simple. Probably because the ground truth does not contain any dynamic objects. Performance for all models is similarly high for the Middlebury Flow data [2]. Most likely because the displacements are very small. A model trained on Sintel [3] shows high compatibility with many diverse data sets.

Our overall observation is that domain similarity for matching is mostly defined by the displacement characteristics and camera hardware, and less by the scenario or realism of the data. The Driving data set for example shows a big discrepancy to KITTI in the cross-evaluation, though both contain traffic scenarios. Reversely, Sintel shares neither the realism nor the automotive setting with KITTI, but still demonstrates high compatibility. This observation is in accordance with the results in [18] on displacement statistics for optical flow. We can further confirm this by an additional experiment. The Driving data set comes with two different focal lengths (15 and 35 mm). The two subsets do not differ in anything else. Performing a cross-evaluation with models trained on KITTI and both versions of Driving, there is a significant loss in domain similarity when switching to the 35 mm focal length, which is also further away from the KITTI camera parameters. Moreover, transfer between Driving with different focal length does also not work very well.

Color. Two questions of interest regarding color spaces are 1.) Which color space provides good generalization properties? and 2.) Does color influence domain adaption? We investigate the first question by training a model on one data set with two different color spaces (RGB and YUV) and evaluating them on the other data sets (each in the respective color space). In our experiments, there is no clear sign that one of the two color spaces should be preferred over the other in terms of generalization. Both models perform similarly on all test data sets. There is also no sign that YUV or RGB color promote the training process. To answer the second question, we train two models on KITTI, one with the original RGB color and one with gray scale converted images to match the color space of HD1K. Intuitively, one would assume the gray scale model to perform better when evaluated on a gray scale data set like HD1K. Contrary, the result of our experiment showed that the color model achieves a higher accuracy on HD1K data compared to the gray scale model. However, when swapping training and evaluation data, a model trained on HD1K performed better on KITTI if the images were converted to gray scale.

Scale. In a similar fashion, the influence of scale spaces was studied. HD1K images have much higher resolution compared to KITTI (cf. Table 1), thus the receptive field of the SDC network (81 × 81 pixels) covers a much smaller part of the visible scene; even more so because the field of view (FOV) of the camera device is smaller (69 ° instead of 90 °). Again, the assumption is that shifting the scale for the training domain towards the scale of the target domain, would improve the transfer. Once more, in contrast to our expectation, a model trained on down-scaled HD1K data did not perform better on KITTI compared to a model trained on full resolution images. Here, the inverse experiment (KITTI model evaluated on full resolution and down scaled HD1K data) indicates also that images should not be scaled to achieve better domain transfer. This might be due to artifacts introduced by the scaling.

Nonetheless, scale is important for detection and matching. The SDC network is specifically designed to deal with varying scales through the use of parallel convolutions with different dilation rates [27]. Table 3 shows some baseline descriptors, the original SDC network, and a multi-scale model, all evaluated on multiple scales of the KITTI data. The heuristic descriptors (SIFT [17], DAISY [30], BRIEF [4]) show an almost quadratic loss in performance when image size decreases, even if they are supposed to be scale invariant. For increased image scale, they perform better. Presumably because smaller patches show fewer deformations, or other variations between images. The implicit multi-scale design of SDC performs extremely well on different scales, with only a small drop in accuracy. For SDC, the performance drops also when the input is upsampled. This is not surprising since the dilation rates can only simulate smaller scales. A model explicitly trained on multi-scale data amplifies the scale invariance even more, showing almost no degradation of the accuracy when the scale changes.

Normalization. Standardization of the input is useful to remove any bias from the data and to scale features into equal range, making them equally important for training. A common practice is to remove the mean pixel value and to scale them so that all channels have unit variance. Surprisingly, standardization is not crucial to train the SDC network. A model trained on normalized images performs as well as a model trained on the original image data.

Table 3: Multi-scale behavior for different descriptors.

Descriptor	×2	Original	×0.5	×0.25
Multi-scale	**96.60**	**97.30**	**96.85**	**96.60**
SDC [27]	94.55	97.25	96.70	93.90
BRIEF [4]	95.00	94.00	90.50	82.15
DAISY [30]	92.80	91.25	88.15	80.80
SIFT [17]	93.90	89.90	81.95	73.65

Anyway, normalization might also be useful to boost transfer learning by adjusting the pixel distribution to better fit the target domain. This, of course, is only possible if imagery for the target is available at training time. When training on a single domain, experiments showed that neither normalization nor a distribution shift help to better generalize to unseen domains. Yet, when training with a mixture of data (as done in the original SDC network), standardization for each training data set separately improves the performance on unseen domains if the test data is also standardized according to its own statistics. For training on multiple domains, a unified normalization based on the pixel distribution of the entire data works also very well and is favorable if a single, unified model for different domains is required.

5. Conclusion

SDC is a neural network architecture with favorable properties for feature description. The implicit multi-scale design, emulated by parallel dilated convolutions, leads to superior matching performance and great invariance to changes of scale. The analysis of the network and its feature representation brought insights on the weaknesses of SDC features which motivated our adjustments of the training schedule. Proper weight initialization and hard mining in the loss computation improved the accuracy and speeded up training by a factor of about 4. More balanced region sampling during generation of training data or learning rate disruption could not improve the networks performance. The evaluation of similarity for different domains gave useful directions to improve the process of domain adaption and the training on multiple data sets. We did also investigate the influence of color, scale, and normalization. The excellent scale invariance of SDC was boosted even more by dedicated multi-scale training. For future work, we are interested in improving feature description to make use of all feature dimensions and to explicitly model a measure of uncertainty or matching likelihood of image points.

References

[1] Christian Bailer, Kiran Varanasi, and Didier Stricker. CNN-based patch matching for optical flow with thresholded hinge embedding loss. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3

[2] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision (IJCV)*, 2011. 2, 3, 6

[3] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, 2012. 2, 3, 6

[4] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *European Conference on Computer Vision (ECCV)*, 2010. 7

[5] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). *arXiv preprint arXiv:1511.07289*, 2015. 6

[6] Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. In *International Conference on Computer Vision (ICCV)*, 2017. 5

[7] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2

[9] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Asian Conference on Computer Vision (ACCV)*. 2010. 3

[10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010. 6

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision (CVPR)*, 2015. 6

[12] Heiko Hirschmüller. Stereo processing by semiglobal matching and mutual information. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2008. 3

[13] Yinlin Hu, Rui Song, and Yunsong Li. Efficient coarse-to-fine patchmatch for large displacement optical flow. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[14] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations (ICLR)*, 2015. 3

[16] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Gussefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, et al. The HCI benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2016. 2, 3, 6

[17] David G Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, 1999. 3, 7

[18] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision (IJCV)*, 2018. 2, 6

[19] Nikolaus Mayer, Eddy Ilg, Philip Hausser, 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 6

[20] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 3, 6

[21] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013. 6

[22] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)*, 2002. 2, 3, 6

[23] Thomas Schöps, Johannes L Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3, 6

[24] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5

[25] René Schuster, Christian Bailer, Oliver Wasenmüller, and Didier Stricker. FlowFields++: Accurate optical flow correspondences meet robust interpolation. In *International Conference on Image Processing (ICIP)*, 2018. 3

[26] René Schuster, Oliver Wasenmüller, Georg Kuschk, Christian Bailer, and Didier Stricker. SceneFlowFields: Dense interpolation of sparse scene flow correspondences. In *Winter Conference on Applications of Computer Vision (WACV)*, 2018. 3

[27] René Schuster, Oliver Wasenmüller, Christian Unger, and Didier Stricker. SDC - Stacked dilated convolution: A unified descriptor network for dense matching tasks. In *Confer-*

ence on Computer Vision and Pattern Recognition (CVPR), 2019. 1, 2, 3, 4, 5, 6, 7

[28] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of CNNs for optical flow estimation. *arXiv preprint arXiv:1809.05571*, 2018. 2, 6

[29] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[30] Engin Tola, Vincent Lepetit, and Pascal Fua. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2010. 7