Deformable Convolution and Optical Flow – Final Report

John Y. Shin
jys2133@columbia.edu

Nicholas Sparks
ns3284@columbia.edu

Jacob Portes
jpp2139@columbia.edu

Abstract

Deformable convolution augments traditional convolution by learning the geometric spatial structure instead of assuming fixed regular grids. We propose to use this module (Dai et al. [2017]) on the computer vision task of **optical flow**, or the distribution of apparent velocities of objects between two or more frames of an image. We added these modules to the FlowNet (Dosovitskiy et al. [2015]) and PWC-Net (Sun et al. [2018]) architectures, and train and test on the MPI-Sintel flow datasets (Butler et al. [2012]). Our preliminary results indicate that deformable convolutions can sometimes improve performance on optical flow tasks.

1 Introduction and Related Work

Convolutional Neural Networks (CNNs) have been incredibly successful at wide range machine learning and computer vision tasks. In particular, CNNs have strong constraints imposed on the network connectivity and the network weights; units are only locally connected in a manner mathematically equivalent to rigid spatial convolution, and these convolutional units share identical weights. This results in far better performance than under-constrained vanilla Fully Connected (FC) neural networks applied to similar tasks (Goodfellow et al. [2016]). However, despite the popularity and success of CNNs with these canonical components, there is no reason *a priori* that the convolutional connectivity should be limited to a regular grid.

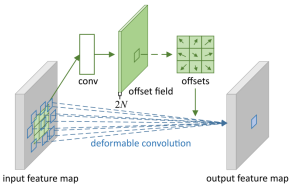


Figure 1: Deformable convolution from Dai et al. [2017]

Deformable Convolutional Networks (DCNs) were first introduced in Dai et al. [2017], with follow up work in Zhu et al. [2018]. One of their key insights was that *relaxing* the spatial connectivity constraint on the convolutional filters could actually improve object detection and segmentation. A natural question to ask is what other computer vision tasks would benefit from this particular form of convolution? We propose that one well suited task is **optical flow estimation**.

Optical Flow is defined as the pattern of apparent motion of objects in a visual sequence, and optical flow fields are used in computer vision tasks such as motion detection, object segmentation, motion alignment, etc. An optical flow field consists of 2D vectors indicating the displacement of points from one frame to the next. Traditional algorithms to estimate local optical flow (e.g. Lucas-Kanade, Horn-Schunk) are based on calculating local spatial derivatives and are not learning-based. CNN-based approaches such as FlowNet from Dosovitskiy et al. [2015], FlowNet2 from Ilg et al. [2017] and PWC-Net Sun et al. [2018] have been applied to optical flow tasks quite successfully, and are quickly replacing these classical algorithms.

We primarily use the MPI-Sintel dataset for training and testing (Butler et al. [2012]). This dataset is derived from an open-source 3D animated short film, and contains ground truth optical flow stored as

Preprint. Work in progress.

horizontal and vertical displacement matrices. The input are two consecutive RGB image frames stored as `.ppm` files, and the target output is the horizontal and vertical optical flow matrices stored as `.flo` files. There are three ways the data is presented for training and testing - "albedo," "clean," and "final." The albedo and clean passes exclude motion blur, camera noise, and other effects which make optical flow estimation difficult, while the final pass includes all of these effects.

We proposed that this combined implementation of DCNs and FlowNet would improve the performance of FlowNet. The most straightforward evaluation criterion is the average endpoint error (EPE) across all frames on Sintel train and test sets. These values are reported on the Sintel website `http://sintel.is.tue.mpg.de/results`.

2 Network Architectures

2.1 FlowNet

FlowNet was the first deep learning approach to optical flow estimation; it was trained on pairs of images with ground truth optical flow fields and consisted of a convolutional downsampling stage and a "refinement" upsampling stage, similar to that of the U-Net architecture (from the same Freiburg group - Ronneberger et al. [2015]). The U-Net architecture was originally designed to take advantage of data augmentation (via contracting and expanding paths) in domains where thousands of annotated training samples were not available, such as in medical imaging. This applies to the field of optical flow estimation as well, as ground truth optical flow is very difficult to obtain outside of synthetic contexts. The Sintel dataset, for example, only consists of a few thousand image pairs for training. However, as they admit in the paper, FlowNet was more of a "proof of concept" and was not competitive with other non-learning-based methods with regards to accuracy. With regarsd to real-time performance, pretrained FlowNet was significantly faster than the computationally expensive non-end-to-end methods.

This work was followed by FlowNet2 by Ilg et al. [2017], which relied on an architecture of stacked FlowNet modules for small displacement (FlowNetSD) and large displacement (FlowNetS and FlowNetC). FlowNet2 finally made end-to-end learning of optical flow competitive and performed on par with state-of-the art methods at the time of its publication in December 2016.

We applied deformable convolution modules to FlowNet2 code from NVIDIA and compared performance.

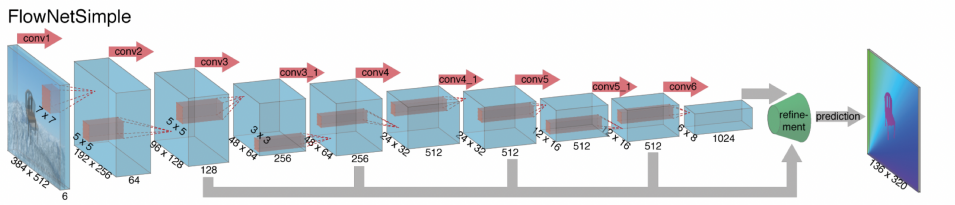


Figure 2: FlowNetSimple (FlowNetS) from Dosovitskiy et al. [2015]. Input consists of two stacked images of size $384 \times 512 \times 3$ each. Only half of the network is shown - the "refinement stage" upscales in order to generate an optical flow field at the resolution of the input images

2.2 PWC-Net

PWC-Net, detailed in Sun et al. [2018], is one of the top performing optical flow networks on both the KITTI and Sintel testing set benchmarks. The architecture of PWC-Net is motivated by well-established principles in computer vision: pyramidal processing, warping, and the use of a cost volume. Pyramidal processing involves applying repeated smoothing and subsampling to an image and is used heavily in applications such as image compression. A cost volume stores the data matching distance-cost for associating a pixel with its corresponding pixels at the next frame; cost volumes are used in standard stereo-matching algorithms (which are special cases of optical flow). PWC-Net also takes advantage of dilated (not deformable) convolution layers that have been shown to improve performance based on contextual information (dilated convolution is a form of rigid upsampling).

The pyramidal processing is learned, as well as the optical flow estimation layer and the context layer. The warping and cost volume layers also do not have any learnable parameters, which reduces the model size. PWC-Net is 17 times smaller in size than FlowNet2.

We applied DCN modules to PWC-Net and tested it's performance.

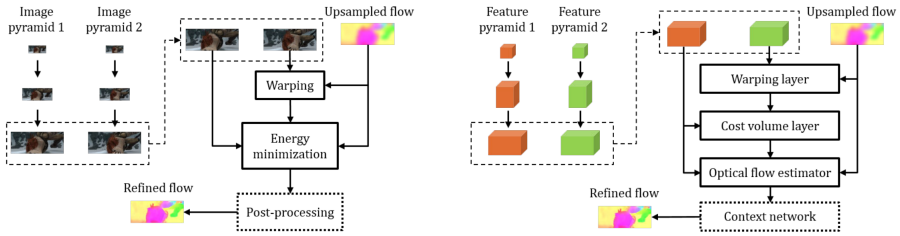


Figure 3: Traditional coarse-to-fine approach (left) vs. PWC-Net (right). PWC-Net is inspired by traditional computer vision techniques such as pyramidal processing, warping, and cost volumes. From Sun et al. [2018]

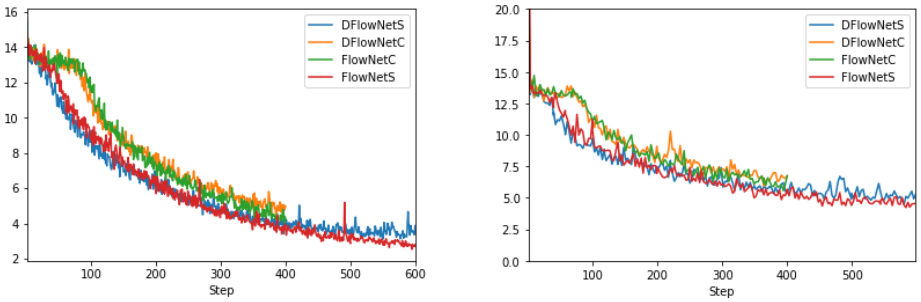
3 Results

We implemented **DCNs** in **FlowNetS** (simple), **FlowNetC** (correlation), and **FlowNetSD** (small displacement) as described in our project proposal. We also implemented DCNs in PWC-Net and in a combined FlowNet2 - PWC-Net stack architecture.

3.1 DCN + FlowNetS on Sintel "final" Dataset

First, we trained on the Sintel "final" dataset, which includes motion blur in the images. For the following set of results, we add deformability in the first two flow prediction layers (`predictflow_6` and `predictflow_5`) out of four flow prediction layers (`predictflow_6`, `predictflow_5`, `predictflow_4`, `predictflow_3`). The learning rate was 1e-4 in all cases, with the ADAM optimizer. The loss function was $L2$.

We compared both the FlowNetS/DFlowNetS architectures, as well as the FlowNetC/DFlowNetC architectures. For the former pair, we trained for 600 epochs, while for the second pair, we train for 400 epochs. The batch size was 8 in all four runs. In the layers where the deformability was turned on/off, the kernel size was 3, with a stride of 1 and padding of 1. In the two loss figures, the red and blue curves indicate FlowNetS/DFlowNetS (simple), and the green and orange curves indicate FlowNetC/DFlowNetC (correlation). For the simple architecture, the deformable version begins with a lower loss rate, but is overtaken by FlowNetS around 400 epochs. For the FlowNetC/DFlowNetC

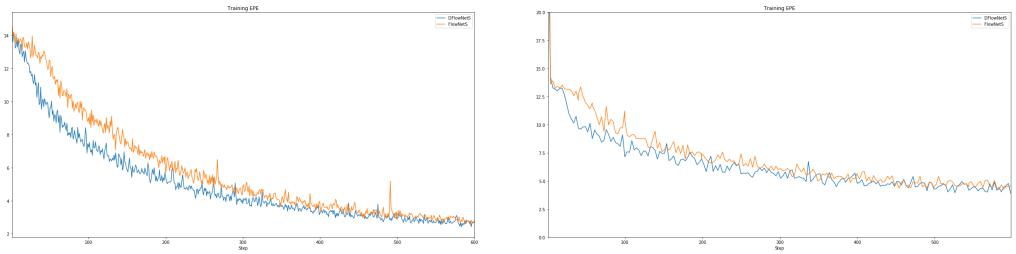


(a) Training (b) Validation

Figure 4: Training and Validation Loss on the Sintel Clean dataset. In both cases, the "vanilla" networks have lower validation loss (L2)

(a) FlowNetS (b) DFlowNetS (c) Ground Truth

Figure 5: Predicted optical flow for the FLowNetS and DFlowNetS architecture. The boundaries for DFlowNet are more diffuse compared to FlowNetS.



(a) Training (b) Validation

Figure 6: Training and Validation Validation EPE comparison between DFlowNetS (blue) and FlowNetS (orange) for 600 epochs. The last prediction layer in FlowNet is turned into a DCN layer for DFlowNetS. The learning rate is 1e-4, the batch size is 8, and the optimizer is ADAM.

architectures, they remain similar for 250 epochs, but FlowNetC has a lower loss afterwards. The validation curves exhibit similar relative behavior.

Next, we compared the predictions of the optical flow for the DFlowNetS and FlowNetS architectures. The prediction with FlowNetS is noticeably crisper, with clearer boundaries between regions. The DFlowNetS prediction is more diffuse, with softer transitions between regions.

Expanding on our work from the midpoint report, one early successful experiment we conducted was a comparison between the original FlowNetS architecture and deformable convolution activated in the last layer. We train on the Sintel final training set, with the training clean pass used as the validation set. We train for 600 epochs, with a learning rate of 1^{-4}, the Adam optimizer, a batch size of 8, and the L2 loss. The deformable version of the network performs better on both the training and validation sets, with a lower average endpoint error (EPE).

3.2 DCNs + PWC-Net

As a follow up, we attempted to conduct an ablation experiment with PWC-Net and deformable layers. Based on the original DCN paper and its comparison of DCNs to atrous convolution, we added deformable layers to the "context layers" in PWC-Net, which use atrous convolution to correct the final flow prediction. We added deformability to the last and third-to-last context layer. We loadws the original pre-trained weights released by the authors of PWC-Net. Interestingly, the authors of PWC-Net claim to have an EPE of 2.31 on the Sintel final training set with these weights, but we did not get the same EPE. We suspect the differences could be due to file I/O and resizing, which they do not clearly write about. We froze the old pre-trained weights and trained the new deformable layers for 20 epochs on the Sintel final training set, with a learning rate of 1e-5, the Adam optimizer, the L2 loss, and a batch size of 1 due to memory limits. We unfroze the weights and continued training for 100 epochs. Both of these networks performed better than FlowNet2, but worse than PWC-Net on the Sintel leaderboards.

We submitted a larger version of FlowNet2 + PWCNet to the MPI-Sintel Website `http://sintel.is.tue.mpg.de/` and ranked 35 overall out of 178. Our submission incoroporated DCNs in the final output layer of the FlowNet2 stack and the final layer of the PWC-Net stack. The various categories are: EPE (Endpoint error over the complete frames), EPE matched (Endpoint error over regions that remain visible in adjacent frames), EPE unmatched (Endpoint error over regions that are visible only in one of two adjacent frames), d0-10 (Endpoint error over regions closer than 10 pixels to the nearest

Rank	Model	EPE all	EPE m	EPE um	d0-10	d10-60	d60-140	s0-10	s10-40	s40+
2	SelFlow	4.262	2.040	22.369	4.083	1.715	1.287	0.582	2.343	27.154
20	PWC-Net	5.042	2.445	26.221	4.636	2.087	1.475	0.799	2.986	31.070
36	F2PD_JJN*	5.530	2.819	27.639	4.539	2.452	2.063	0.918	3.113	34.257
49	FlowNet2	5.739	2.752	30.108	4.818	2.557	1.735	0.959	3.228	35.538

Table 1: Comparison of our FlowNet2+PWC-Net architecturec with PWC-Net, FlowNet2 and SelFlow on the Sintel final dataset

occlusion boundary), d10-60 (Endpoint error over regions between 10 and 60 pixels apart from the nearest occlusion boundary), d60-140 (Endpoint error over regions between 60 and 140 pixels apart from the nearest occlusion boundary), s0-10 (Endpoint error over regions with velocities lower than 10 pixels per frame), s10-40 (Endpoint error over regions with velocities between 10 and 40 pixels per frame), 40+ (Endpoint error over regions with velocities larger than 40 pixels per frame). We consistently outperform FlowNet2 in all categories, and consistently perform worse than the top algorithm SelFlow in all categories (although SelFlow is listed as 2, 1 is Ground Truth).

Figure 7: F2PD_JJN.

Figure 8: SelFlow - MPI-Sintel Leader.

4 Discussion

We have yet to test this on the KITTI datset from Menze et al. [2018], which consists of 200 training scenes and 200 test scenes captured by driving around the city of Karlsruhe, Germany with up to 15 cars and 30 pedestrians visible per image. Ground truth for this dataset is based on a laser scanner and is more sparse and approximate than the Sintel ground truth.

We would ultimately like to apply our network to the video object detection network implemented in *Flow-Guided Feature Aggregation for Video Object* Detectionby Zhu et al. [2017]. This paper implemented the original FlowNet architecture as part of a larger object detection network. If our architecture works as planned we hope to implement it as a module in this larger object detection network.

DCNs were originally shown to enhance performance on object segmentation tasks; for example, when DCNs were added to ResNet-based "DeepLab" Chen et al. [2018] at various final layers, they performed better than DeepLab. Thus we think that optical flow estimates might be improved if objects are segmented early on in the network. However, we do not have any evidence that including DCN modules in the early layers of the network is "segmenting" objects.

We also were hoping that if DCN modules were included in the final predictive stages of the FlowNet architecture, it would form "interpretable" filters corresponding to flow offsets. We have yet to test this.

5 Code

The code is located at the following github:

`https://github.com/johnshin86/flownet2-pytorch`

Our best submission to the MPI-Sintel test set can be found here under the name "F2PD_JJN": `http://sintel.is.tue.mpg.de/results`

A Mathematical Background

A.1 Deformable Convolutions

The two modules outlined in Dai et al. [2017] are *deformable convolutions* and *deformable region of interest pooling*.

Regular 2D convolution samples over a regular grid, R, over the input feature map x. For each location p_0 on the output feature map y, we have the weighted sum:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n) \tag{1}$$

where p_n enumerates the locations in R. Deformable convolutions adds an offset to normal convolutions, Δp_n, where $\{\Delta p_n | n = 1, ..., N\}$, where $N = |R|$.

$$y(p_o) = \sum_{p_n \in R} w(p_n) \cdot x(p_o + p_n + \Delta p_n) \tag{2}$$

The sampling now occurs on the irregular locations $p_n + \Delta p_n$. Since Δp_n is often fractional, $x(x_0 + p_n + \Delta p_n)$ is implemented as $x(p) = \sum_q G(q, p) \cdot x(q)$, where G is given by:

$$G(q, p) = g(q_x, p_x) \cdot g(q_y, p_y), \quad g(a, b) = max(0, 1 - |a - b|) \tag{3}$$

where $p = p_0 + p_n + \Delta p_n$ and q enumerates all integral spatial locations in the feature map x.

References

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

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 *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

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, October 2012.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *arXiv preprint arXiv:1811.11168*, 2018.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017.

O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. URL `http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a`. (available on arXiv:1505.04597 [cs.CV]).

Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.