Machine Perception Report

Project: Learning human optical flow - Group: SpaghettiCode

Antonio Lopardo
alopardo@student.ethz.ch

Luca Malagutti
lmalagutti@student.ethz.ch

Federico Gossi
fgossi@student.ethz.ch

ABSTRACT

The task of optical flow estimation consists in reliably identifying the pixel-wise motion of objects between two consecutive images or frames of a video. Recent approaches to this task employ deep neural architectures extensively trained on multiple synthetic datasets. One of such datasets is the Multi-Human Optical Flow (MHOF) dataset [7]. This report will detail how we have adapted, modified, and fine-tuned the state-of-the-art neural architecture, Recurrent All-Pairs Field Transforms (RAFT) [9] to a subset of the MHOF dataset. Our main contributions are the addition of a form of teacher-forcing to the initialization of the recurrent submodule of the architecture, a more extensive use of channel dropout on convolutional layers to avoid overfitting on our small provided dataset and the generation of new synthetic data. The final submission, which combines all these three contributions, achieves an EPE of 0.345 as the public test score of ETH's Machine Perception Project 6 competition.

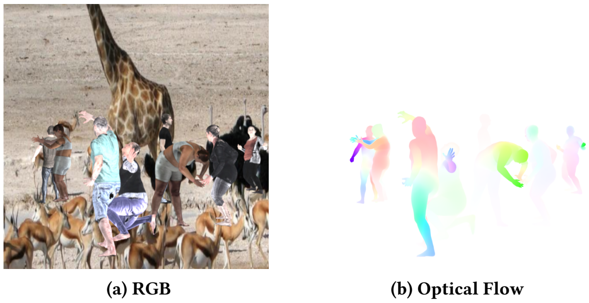
1 INTRODUCTION

Optical flow estimation has many application domains [10], such as Visual Surveillance, Robot Navigation and Image Interpolation. Detecting optical flow differs from other computer vision tasks like image classification or object detection in the difficulty of efficient dataset annotation. The complexity of comparing two images pixel by pixel makes creating real-world datasets especially challenging for human annotators. As a result, most of the datasets used for evaluating models for optical flow are computer-generated. These datasets [2, 4, 6] vary in size, complexity and subject, but most try to test models on their ability to deal with occlusions, when objects in movement overlap or cover one another between two frames, and to handle perturbations both of the images themselves and of the camera viewpoint, to make testing scenarios more realistic.

MHOF focuses on both these issues by including in its rendered scenes multiple human subjects, up to eight, at different depth levels and by adding motion blur and Gaussian blur to the images as well as camera motion to 30% of the sequences. The scenes' backgrounds are also quite varied and include both outdoor and indoor pictures.

RAFT[9] was not pretrained on MHOF [7], but comparing it with PWC-Net [8] on the Sintel [2] and KITTI [4] benchmarks it displayed clear advantages. Thus, after experimenting with PWC-Net, we worked on fine-tuning RAFT on MHOF with similar learning schedules to those described in its paper[9].

Our final model includes three main contributions: teacher forcing, channel dropout and new data generation. More specifically, for each pair of images fed to the network, we initialize the first step of RAFT's recurrent layers with the ground-truth flow of the previous image pair in the sequence. We also use substantial levels of dropout to increase regularization by randomly dropping some channels in the convolutional layers of the architecture. 4127 new image pairs were also generated in an effort to reduce overfitting. To generate these new samples, we used the code available at [1].


(a) RGB

(b) Optical Flow

Figure 1: Image from the MHOF dataset with its corresponding optical flow to the next image in the sequence

While training RAFT's original architecture, we also tried to adapt the final upsampling layer to make it more suited to our task and attempted to modify the fine-tuning schedule by training the model on simpler examples in the beginning.

2 RELATED WORK

The architectures that we experimented with, PWC-Net[8] and RAFT[9], share a similar approach to their first feature extracting modules where they both use stacked convolution layers applied to both images that culminate in pixel by pixel correlation layer, whose output is used as input to the submodule of the network that produces the predicted flow. However, they differ most in how they generate and iteratively improve the optical flow. PWC-Net uses a coarse-to-fine approach to reduce large pixel by pixel displacements between the images. An initial low-resolution version of the predicted flow is improved several times at different resolutions to finally produce the full-size optical flow. In contrast, RAFT refines its predicted low-resolution flow with repeated iterations through recurrent GRU units, without employing the coarse-to-fine pyramidal approach employed by most previous state-of-the-art models. Despite having fewer parameters, RAFT performs better than PWC-Net on the Sintel [2] and KITTI [4] benchmarks, therefore we decided to improve on it during the competition.

3 METHOD

Starting from a model checkpoint pretrained on the FlyingChairs [3], FlyingThings [6], Sintel [2] and KITTI [4] datasets made available by RAFT's authors at [2], we added three distinct contributions to the given architecture, improving the base model by 15.4% when fine-tuning RAFT on the MHOF [7] dataset. We elaborate more on our additions in more detail in the three subsections below.

[1] https://github.com/DavHoffmann/Multi-humanDataGeneration

3.1 Teacher Forcing and warm-start

As discussed in [9], warm-start refers to the initialization of the first state of the iterative flow generating sub-module with the optical flow prediction of the previous pair of frames, instead of using an initial flow field set to a tensor of all zeros. The original RAFT architecture already used warm-start when reporting its final testing results, but it only enabled warm-start during testing and evaluation on the final dataset, Sintel [2].

To improve the learning accuracy, we add a variant of warm-start to the training procedure of the model, while also enabling RAFT's original warm-start technique during evaluation and testing when fine-tuning on MHOF [7]. We refer to warm-start during training as *"Teacher Forcing"*. In Teacher Forcing, we initialize the flow field to the downsampled version of the previous ground truth flow immediately preceding the current pair in the sequence. At the beginning of each sequence, we initialize the flow state to zero, as it was originally done.

The reasoning behind our addition of Teacher Forcing was two-fold: firstly we knew that warm-start obtained better results on the Sintel dataset, so it made sense to also enable it when fine-tuning on our dataset, secondly, we hypothesized that the model would find it difficult to generalize optical flow prediction when starting from a meaningful initialization during inference if it was only trained when starting from an empty flow field. Therefore, we tried to improve the generalization capabilities of the model by initializing the flow field to a meaningful representation, such as the ground truth of the previous pair of frames, even during training.

In our implementation, to associate more easily the current pair of images to its immediately preceding ground truth flow, we changed the batch size of the model to 1 and disabled data sample shuffling between epochs. To make the ground truths of the previous frame as similar as possible to the prediction that has to be computed by the model, we also toned down the default spatial and stretching image and flow transformations, and also completely disabled the random horizontal and vertical flip transformations, which were originally used by RAFT for data augmentation. We have also removed PyTorch's ColorJitter photometric transformation of the input images obtaining better results.

3.2 Dropout regularization

We decide to use dropout due to the small size of the provided dataset, and the addition of teacher forcing. We experimented with dropout probabilities ranging from as low as 0.05 to as high as 0.5 but finding 0.4 to be best. It is therefore possible then that the model might have benefited from less regularization thanks to the addition of new training data. It is relevant to note that the authors of RAFT pointed out that their model, thanks to a smaller number of learnable parameters than previous state-of-the-art architectures, was less prone to overfitting allowing them to disable the dropout layers completely. Thus, our results show just how much our final model relies on the previous flow introduced to it with teacher forcing and the fine-tuning of MHOF for its prediction on the test set.

Table 1: Result of the ablation study of the contributions of the report.

Method	Validation	Public Test	Improvement
RAFT@75K	0.2123	0.4079	-
RAFT+TF	0.1721	0.3639	10.8%
RAFT+TF+DP	0.1621	0.3498	14.2%
RAFT+TF+DP+EX	**0.1583**	0.3494	14.3%
RAFT+TF+lowDP+EX	0.1595	**0.3452**	15.4%

3.3 New data samples generation

The code[1] needed to generate new synthetic sequences was made available by the authors of the MHOF dataset as part of the release of the original implementation in 2018. As of November 25, 2021however working with some of the dependencies that they originally used is not always straightforward since some rely on deprecated standards and versions which are not easily supported anymore. After some troubleshooting, we managed to run their code without errors and produce 4127 new images. We ran the script to generate new sequences intermittently for close to 10 days on an Ubuntu 18.04-based machine running an Intel i5-4670k and 24GB of ram. We manually verified that our generated sequences did not contain artifacts or missing frames and integrated them with the training data initially provided for the competition.

The first experiments in which we used the generated sequences also involved teacher forcing and a dropout probability of 0.4. In these settings, the new data did not provide a significant improvement, however, changing the dropout probability to 0.2, helped us reach our best public test score of 0.345.

4 EVALUATION

We perform an ablation study on our main contributions: in Table 1 TF refers to teacher forcing, DP to channel dropout with probability at 0.4, lowDP to channel dropout with probability 0.2, and EX to the addition of 4127 new images to the training set. Validation and Public Test refer to the EPE scores (lower is better) on the provided validation set and on the public leaderboard test set, respectively. The improvement is relative to the Public Test score of RAFT@75K. RAFT@75K refers to the original architecture fine-tuned on MHOF [7] for 75K iterations with batch size 2. We also kept the original learning rate, weight decay and gamma hyperparameters as in [2]. All models with teacher forcing use batch size 1, for a fair comparison we trained them for 150K iterations, leaving other hyperparameters unchanged. RAFT+TF+lowDP+EX is our final submission to the project leaderboard.

5 DISCUSSION

The three main contributions discussed in the Method section were not the only novel ideas we tried to apply to the task of detecting multi-human optical flow. The other major experiments we ran involved both significant changes to the architecture and a new training schedule.

In this section we will discuss these experiments, introduce our

[2] https://github.com/princeton-vl/RAFT

Table 2: Results of the experiments described in the Evaluation section.

Method	Validation	Public Test	Improvement
RAFT Baseline	**0.2163**	**0.4087**	-
RAFT + UpSample	0.2230	0.4282	-4.76%
RAFT + CurrLearning	0.2285	0.4229	-3.74%

reasoning for trying them and speculate on why they did not work when compared to a baseline consisting of a RAFT model fine-tuned for 50K iterations on MHOF [7] without any architectural or hyperparameter changes. The results of these experiments are summarized in Table 2.

5.1 Curriculum Learning

As shown already in [1] and more recently for Deep Neural Networks applied to computer vision in [5], designing the training schedule of a model in order to let it train on easier samples first can increase learning speed and improve final performance. We tried to use this principle to train a model to detect human optical flow since we noticed it applied to the training schedule followed by many start-of-the-art models, including RAFT. More specifically, as detailed in its paper, RAFT is first trained on the simpler FlyingChairs and FlyingThings3D for 100K iterations each and only later on the more complex Sintel, KITTI-2015, and HD1K datasets. We noticed significant differences in the EPE loss values of different training samples fed to a pretrained RAFT model. Therefore, we attempted to "bootstrap" a partition of the training set in four equally-sized training bins split at the 25th, 50th and 75th percentile of the EPE loss of each sample. After bootstrapping the training bins, we consecutively fine-tuned RAFT for 12,5k iterations on each bin to make it comparable to the RAFT@50K baseline we mentioned at the beginning of this section. However, as shown in Table 2 the model fine-tuned with this training schedule performed worse than the aforementioned baseline.

Looking at validation scores reached by these models during training gave us a clue as to why this technique was not working. They made quick progress when training on the first bin, getting lower validation scores than the baseline at the same iterations, but as the models switched training bins the validation scores first deteriorated significantly and then went back down to the levels reached at the end of training on the first bin. This instability undoubtedly hurts the models, none of which could surpass the baseline, despite several attempts at changing hyperparameters and/or scheduling details.

5.2 Modified upsampling layer

RAFT internally produces optical flow predictions at 1/8th of the size of its input images. These intermediate flow predictions are then upsampled at the original input size using two convolutional layers to obtain a full resolution optical flow by taking each full resolution pixel as the convex combination of its 9 neighbors at coarse resolution. Since the submission server expects flow predictions at 1/4th of the image input size, we initially decided to manually downsample the full-sized outputs of RAFT at 1/4th of their size before uploading them to the scoring server. To avoid what we thought was an unnecessary downsampling operation, we tried to change the last layer of the custom upsampling submodule of RAFT to upsample only to 1/4th of the input size, as opposed to a full resolution that we could not submit without further processing. In our experiments, however, we discovered that replacing the originally trained layer with a randomly initialized layer, and then fine-tuning it on MHOF [7], performed worse than simply applying an additional downsampling operation on RAFT's original output.

This could be caused by the fact that our dataset is too small to properly perform transfer learning, especially given how much bigger were the datasets involved in the training procedure with the original layer. The scarce amount of available training data made the new layer unable to adapt itself to the signals of the preceding layers. After fine-tuning, using our custom smaller upsampling layer led to suboptimal predictions, performing 4.76% worse than the baseline.

5.3 New Architecture

Since the task of the MP-Project6 competition does not involve any assessment of real-time performance and the RAFT architecture is smaller and therefore less prone to overfitting compared with previous state-of-the-art models, we looked into increasing the modeling power of RAFT by adding more layers. In the feature extractor, we added a single residual unit to each residual pair as well as a new group of three residual units between the last two triples. After the correlation and flow features layers, we introduced a new convolutional layer each, and we did the same for the section of the network that outputs the $\Delta Flow_t$.

Making these changes to the architecture required several adjustments to existing layers that made it impossible to load in the model the pre-trained weights made available by RAFT's authors. Therefore, we attempted to fully retrain this larger version of the architecture, that counted 8.9M parameters, from scratch, following the training schedule detailed in RAFT's own code repository. We started training on the FlyingChairs [3] dataset. However, due to hardware limitations on the batch size, the model failed to converge in a reasonable time. Another issue we encountered in retraining the architecture lied in the size of the other datasets, as some, like FlyingThings3D, exceed 600 GB, a dimension unmanageable for our means. After trying unsuccessfully to continue training the model on subsets of the data, we decided to abandon this attempt, however, we still believe that, given adequate processing power and loosening the constraint of performing optical flow estimation in real-time, a larger model, when trained successfully, could perform better than the original version RAFT on this specific task.

6 CONCLUSION

In this report we have outlined the three main contributions and some of the experiments that allowed us to take the existing state-of-the-art architecture for optical flow estimation and improve its performance after fine-tuning on a subset of the MHOF [7] dataset. We obtain an EPE of 0.345 as our public test score, which, as of November 25, 2021, places us in the first position on the public leaderboard for the "Learning human optical flow" competition of the Machine Perception course.

REFERENCES

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. Association for Computing Machinery, New York, NY, USA, 41–48. https://doi.org/10.1145/1553374.1553380

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. 2012. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV) (Part IV, LNCS 7577)*, A. Fitzgibbon et al. (Eds.) (Ed.). Springer-Verlag, 611–625.

[3] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. 2015. FlowNet: Learning Optical Flow with Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV)*. http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[5] Guy Hacohen and Daphna Weinshall. 2019. On The Power of Curriculum Learning in Training Deep Networks. arXiv:cs.LG/1904.03626

[6] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. 2016. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. http://lmb.informatik.uni-freiburg.de/Publications/2016/MIFDB16 arXiv:1512.02134.

[7] Anurag Ranjan, David T. Hoffmann, Dimitrios Tzionas, Siyu Tang, Javier Romero, and Michael J. Black. 2020. Learning Multi-human Optical Flow. *International Journal of Computer Vision* 128, 4 (Jan 2020), 873–890. https://doi.org/10.1007/s11263-019-01279-w

[8] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. 2018. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. arXiv:cs.CV/1709.02371

[9] Zachary Teed and Jia Deng. 2020. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. arXiv:cs.CV/2003.12039

[10] Zhigang Tu, W. Xie, Dejun Zhang, R. Poppe, R. Veltkamp, Baoxin Li, and Junsong Yuan. 2019. A survey of variational and CNN-based optical flow techniques. *Signal Process. Image Commun.* 72 (2019), 9–24.