Learning Human Optical Flow (Project 6)

Luke Smith
ETHZ
lusmith@student.ethz.ch

Clément Guerner
ETHZ
cguerner@student.ethz.ch

Baptiste Freydt
ETHZ
bfreydt@student.ethz.ch

ABSTRACT

Human optical flow is a computer vision task consisting of identifying the movement of human bodies across a sequence of images. This task is important because it allows machines to predict human actions and react accordingly. In this report, we explore the effectiveness of fine-tuning a neural network-based architecture to the task of multi-human optical flow. We use a synthetic dataset to train our model, and apply pre- and post-processing techniques to improve performance. While we obtain promising results with this approach, some of our ideas did not bear fruit.

1 INTRODUCTION

Optical flow is the motion (displacement) of all pixels across two images in a sequence. We consider the location of each pixel in the first image, and attempt to match it to a pixel in the second image. The optical flow therefore captures the horizontal and vertical displacement for each pixel in the initial image, resulting in a vector field.

Our task is to learn human optical flow, which is the application of this approach to image sequences containing human bodies in motion. This task is of great interest because it enables machines to identify human movement using only camera input, allowing them to predict which action a human is going to perform. For example, an autonomous car could use optical flow to predict the actions of pedestrians or cyclists.

In order to calculate human optical flow, we use the PWC-Net architecture. PWC-Net relies on several classical approaches to calculating optical flow, including the image pyramid. This pyramid is formed by repeatedly downsampling an image to obtain progressively coarser versions of the image. The optical flow is estimated at each level of the pyramid, facilitating the detection of large motions at coarse levels and small motions at finer levels. PWC-Net's implementation of the pyramid uses downsampled image features generated by a CNN instead of the image itself at each level of the pyramid, as shown in Figure 1. [4] This technique inspired us to try different up and downsampling approaches as documented in the Method section.

PWC-Net reports good performance in predicting large motions and areas away from motion boundaries, but it struggles with small objects moving rapidly such as arms [4]. Our error analysis, presented in Method 2.1, shows similar results.

We leverage the Multi Human Optical Flow (MHOF) dataset from Ranjan et al. to train and test our model [2]. The data are generated as follows: a scene is created using a high quality image of an indoor or outdoor space as the background, and humans generated using the parametric human body model SMPL+H are then added to the scene. This particular model includes articulated fingers to be more realistic [2]. See Figure 2 for an example of an image generated using this approach, with the corresponding flow.

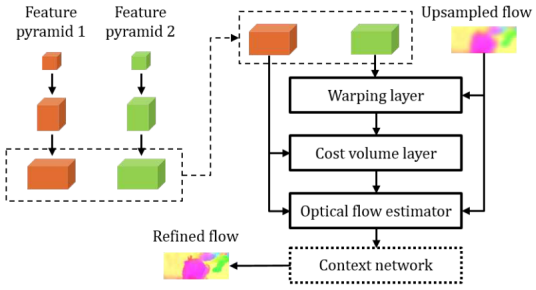


Figure 1: PWC-Net architecture

Our evaluation metric for optical flow predictions is the end point error (EPE), which measures the euclidean distance between the ground truth vector and the predicted vector for every pixel.

As a starting point for the task, we were given a PWC-Net pretrained for the task of MHOF. Our method, as outlined below, consists of further fine-tuning the PWC-Net on the MHOF dataset so as to refine the model to our specific task. We then added some pre- and post-processing steps using segmentation masks to address patterns found in our prediction errors. Through our experiments, we found that downsampling the input image and correcting background pixels led to the best performance.

Figure 2: MHOF image example

2 METHOD

2.1 Error Analysis

We started with a qualitative analysis of our prediction error: using the provided model, we generated predictions on the validation set and produced a heat map of EPE between the predicted and ground truth optical flows for each validation sample. An example of a heat map can be seen in Figure 3.

The heat maps showed that the hands, arms, legs and feet were the main sources of errors across samples. We also noted that the

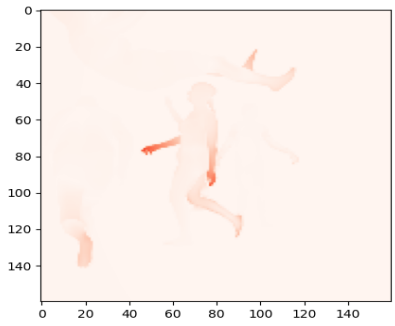


Figure 3: Qualitative evaluation: EPE Heat Map

edges of the legs and arms showed higher EPE values than the center of these body parts in some cases. Together, these observations corroborated the results reported by PWC-Net discussed earlier. Finally, for certain samples, the background of the image was an important source of EPE.

2.2 Model architecture

Next, we sought to alter the PWC-Net architecture to address these issues. For example, we considered modifying the horizontal and vertical downsampling rates of the image pyramid in the model to bring into focus arm and leg movements. However, we realized that because of the size of the model and the computational resources at our disposal, making such alterations to model parameters and architecture would not be possible. A single pass on the train subset of MHOF provided for the project took approximately 8 hours.

2.3 Data generation and training schedule

Ilg et al. explain that the training schedule, and specifically training on simple image sequences in early epochs, improves their model's performance [1]. Although we were not allowed to download MHOF, we were allowed to generate it ourselves (the code from the paper is freely available) [2]. Using this library, we attempted to generate simpler MHOF samples by changing the parameters governing MHOF data generation. Our hypothesis was that using this simpler generated data at the start of the fine tuning process would both improve performance and help avoid overfitting on the provided data.

Unfortunately, after solving many installation issues due to the very poor quality of the repository, we realised that it was not possible to generate MHOF on Leonhard due to compute capacity constraints. We hit writing restrictions on the scratch folder after 4 hours of running time, having only managed to generate 4GB of data out of 300GB in that time.

2.4 Fine tuning

Sun et al. found that training PWC-Net on test data led to good overfitting results for a specific application [4]. The pretrained model provided to us was trained on a variety of data from different data sets in order to generalize well. By fine tuning it on the MHOF dataset, we hoped to improve model performance on this dataset specifically. We used the same training procedure as the authors and used the Adam optimization algorithm.

2.5 Pre-processing

Inspired by the idea of the image pyramid, which downsamples the image features to different scales to enable the detection of large and small motions, we tried four different dimensions for inputting images into the model at prediction time: (384, 384, 3), (448, 448, 3), (512, 512, 3) and the original size, (640, 640, 3).

2.6 Post-processing and segmentation mask

Drawing from observations made using the EPE heatmaps, we used the segmentation masks provided with the data to attempt to address two recurring types of error: first, error in background pixels far away from human bodies, and second, errors at the boundaries between an arm and the background, for example.

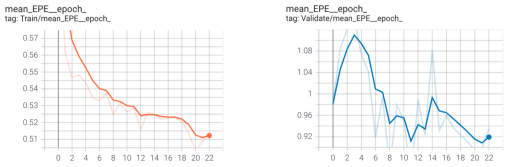
To address the background errors, we used the mask to obtain the predicted flow for each pixel of the background, took the median and then set every background pixel to this value. Our hypothesis was that this would help alleviate both background and boundary errors, since background movement appeared to be uniform and the background pixels at the boundaries would be corrected.

For the interior of the body parts, we also attempted to smooth the flow inside the body part using a Gaussian blur. Given that for some samples, pixels at the center of the body part had less error than those on the boundary, we hypothesized that this filter would reduce boundary errors.

3 EVALUATION

3.1 Training

We trained the model on the provided train data set for 22 epochs (each epoch took an hour). We obtained significant improvements in performance in both training and validation set EPE, as shown in Figure 4.

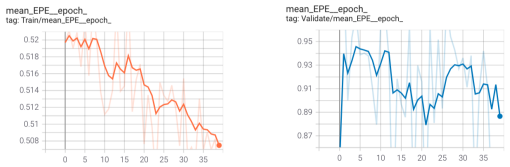


(a) Training set **(b) Validation set**

Figure 4: Fine Tuning First Run

There was little evidence of overfitting since both the train and validation set EPE were decreasing, so we decided to train for another 40 epochs. This second training is documented in Figure 5. This time, the model appears to be overfitting, since the validation EPE is relatively constant on average, with high variance across epochs (light blue line in the graph). We deemed that further training on the train set would only lead to more overfitting, so we stopped here.

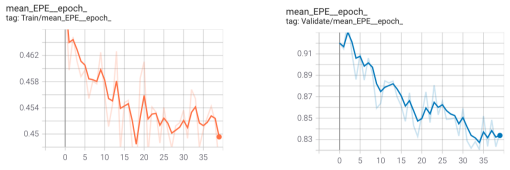
For the final submission, we trained for a further 40 epochs on the combined train and validation set so as to leverage all available



(a) Training set (b) Validation set

Figure 5: Fine Tuning Second Run

data. Figure 6 documents EPE for train and validation sets, both of which were used for training in this final run. Since this training increased train set performance and led to a better result on the public leader board, we chose this submission.



(a) Training set (b) Validation set

Figure 6: Fine Tuning Train+Val Final Run

3.2 Pre and post-processing

Figure 7 shows the results of experiments with pre- and post-processing described in Method 2.5 and 2.6. For each method combination, we report the mean EPE for the entire validation set as provided for the project. From these results, we conclude that:

- Without background correction an input size of 448x448 is the best.
- With background correction the original input size of 640x640 performs better.
- Body part smoothing reduces performance.

We then took the results of this experiment and uploaded two submissions to the public leaderboard, obtaining contradictory results shown in Figure 8. While input size of 640 with background correction performed best on validation, the public test score was significantly lower for 448 + background correction.

4 DISCUSSION

Using the segmentation mask to improve optical flow prediction is only possible due to the fact that the MHOF data set is synthetic. Detecting the contours of objects and identifying the layers of an image are important research areas in optical flow, as shown by Sun et al. [3]. Our approach with segmentation masks aims to give some insight into the performance improvements that could be made with these techniques on MHOF.

The first noteworthy result is that downsampling the input image conflicts with background correction. Downsampling has a smoothing effect, both on the background and the body parts. Meanwhile, background correction addresses two sources of error: non-uniform background flow predictions and errors at motion

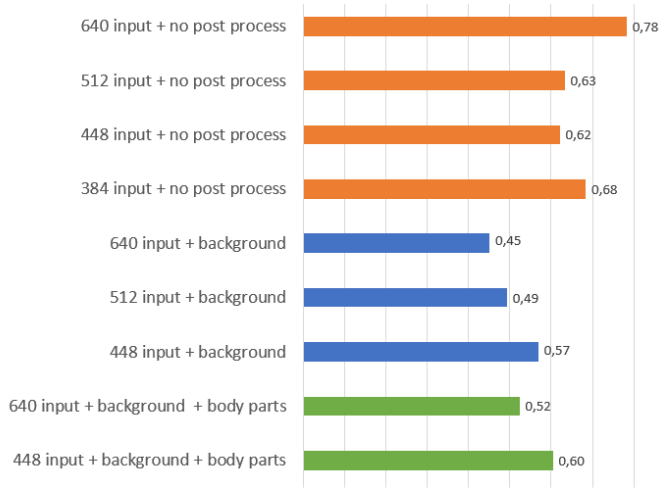


Figure 7: Validation EPE with pre- and post- processing

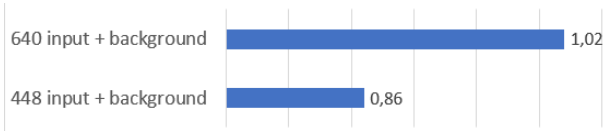


Figure 8: Public test EPE with pre- and post-processing

boundaries. When applied together, the background smoothing gains from downsampling are negated through the post-processing correction. Instead, downsampling increases error by reducing accuracy at and within motion boundaries.

We also note that the training process uses images of size (448x320), and this is the result of cropping the input image rather than downsampling. In this sense, downsampled images at prediction time are less similar to images the model was trained on, which should lower the performance.

However, we obtained opposite results on the public test set, where the best result was obtained by applying both downsampling and background correction. Because the test set comprises only 5 unique scenes, we suspect that the downsampling has a large positive effect on a few of these, while this effect is inverted on average on the much larger validation set.

Finally, we found that smoothing within a body part worsened performance. Looking at Figure 3, we see that only one of the legs shows increased errors at the boundaries, while the hands and arms have more uniform error. In the case of uniform error, body part smoothing can't correct the errors of the model.

5 CONCLUSION

In this project, we experienced some of the challenges of working with compute-intensive neural network-based architectures, and we were unsuccessful in implementing some of our ideas. Yet, our results show that fine-tuning a model for specific task and data set can lead to significant performance gains. By researching classical approaches to optical flow, we were able to develop pre- and post-processing experiments that also improved performance. Finally, we

learned that error visualization is a crucial step towards identifying and fixing the sources of prediction errors.

REFERENCES

[1] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2016. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. arXiv:1612.01925 [cs.CV]

[2] 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

[3] Deqing Sun, Erik B. Sudderth, and Michael J. Black. 2012. Layered segmentation and optical flow estimation over time. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 1768–1775. https://doi.org/10.1109/CVPR.2012.6247873

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