Unsupervised Convolutional Neural Networks for Notion Estimation

Zhang, Liqiang

May 29, 2018

1. Introduction

In this paper, the author propose training a CNN for motion estimation in an unsupervised manner. They do so by designing a cost function that is differentiable with respect to the unknown motion field and, therefore, allows the back-propagation of the error and the end to end training of the CNN. The cost function builds on the widely used optical flow constraint - the major difference to Horn-Schunk based methods is that the cost function is used only during training and without regularization. Once trained, given a pair of frames as input the CNN gives at its output layer an estimate of the motion field. In order to deal with motions large in magnitude, the author embed the proposed network in a classical iterative scheme, in which at the end of each iteration the reference image is warped towards the target image and in a classical coarse-to-fine multiscale framework. They train our CNN using randomly chosen pairs of consecutive frames from UCF101 dataset and test it on both the UCF101 [1] where it performs similarly to the stateof-the-art methods and on the synthetic MPI-Sintel dataset where it outperforms them.

2. Method

At the heart of all motion estimation methods is the minimization of the difference between features extracted at a certain location x, y at the reference frame t and its correspondence in the frame $t + dt$. The now classical Horn and Schunck method penalizes the deviation from the assumption of constant intensity, that states that the intensity at a pixel in the reference frame at time t and the intensity at its correspondence at time $t + dt$ are the same. Formally, the goal is the minimization of the motion compensated intensity differences, that is,

$$\sum_{x,y=1}^{M} |I(x+u(x,y), y+v(x,y), t+\Delta t) - I(x,y,t)|^2 \quad (1)$$

where $I(x, y, t)$ is the intensity at pixel (x, y) at frame t, and $F(x, y) \triangleq [u(x, y), v(x, y)]$ is the unknown motion vector at pixel x, y. Clearly, $u(x, y)$ and $v(x, y)$ are respectively the horizontal and vertical displacement of the pixel with coordinates (x, y). To arrive at a computationally tractable method, Horn and Schunck introduced a regularization term that penalized discontinuities in the motion field and linearized the cost by taking the first order Taylor expansion with respect to the horizontal and vertical displacements. By doing the latter, they arrived at the optical flow equation $uI_x + vI_y + I_t = 0$, where I_x, I_y and I_t are the horizontal, vertical and temporal intensity derivatives respectively, and penalized deviations from it [3]. In the equation above, we omit the pixel coordinates for notation simplicity.

In order to reduce the influence of outliers the author use a robust error norm, that is the Charbonnier penalty $\rho = \sqrt{x^2 + \epsilon}$, a differentiable variant of the $L1$ norm, the most robust convex function. Formally, during training we learn the CNN by optimizing the sum of costs that for a pair of images $I(t)$ and $I(t + dt)$ are defined as follows:

$$E(F) = \sum_{(}^{M} x, y = 1)\sqrt{(uI_x + vI_y + I_t)^2 + \epsilon} \quad (2)$$

where the image coordinates x, y are omitted for notation simplicity.

2.1. Architecture

The author propose a fully convolutional neural network with 12 convolutional layers. The architecture could be imagined as two parts. The CNN makes a compact representation of motion information in the first part which involves 4 downsamplings. This compact representation is then used to reconstruct the motion field in the second part which involves 4 up-samplings. The up-sampled is performed by simply repeating the rows and columns of the feature maps. Since these proposed DNN is fully convolutional, the input could be of any size. Fig. 1 shows the two parts of the proposed CNN. To update the CNN weights during the training phase, the author used ADAM and calculate that spatiotemporal intensity derivatives I_x, I_t and $I_t$as proposed in [2].

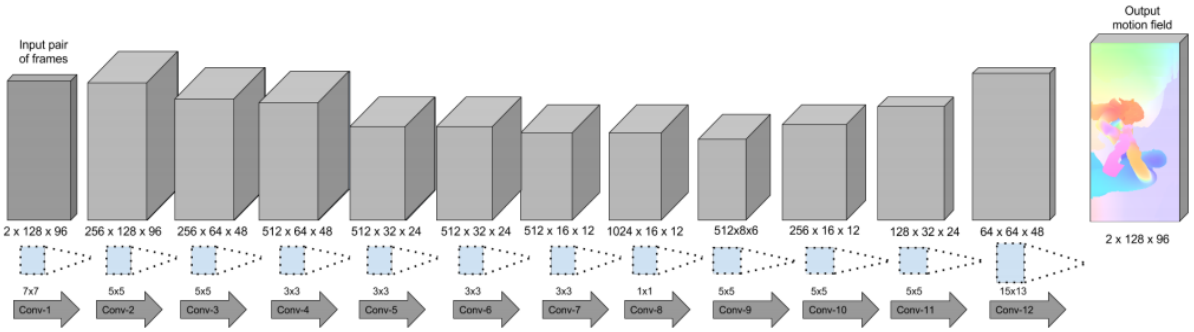

Figure 1. The architecture of our proposed CNN. We have assumed the height and width of the input is 128x96. The illustrated motion field is chosen from MPI-Sintel dataset.

Table 1. Evaluation of different methods on UCF101 dataset. AEE-05 stands for Average End-point Error for the motions smaller than 5 pixels. AEE-5so refers to the motions bigger than 5 pixels, and AEE-tot refers to the total error value. AAE stands for Average Angular Error.

Method	AEE-05	AEE-5so	AEE-tot	AAE-05	AAE-5so	AAE-tot
DeepFlow	0.30	3.99	0.47	9.35	14.65	9.59
HAOF	0.37	4.59	0.56	10.95	19.40	11.33
LDOF	0.35	2.85	0.46	9.91	9.72	9.9
USCNN	0.46	8.7	0.81	12.74	59.50	14.70

3. Experiments

Table 1 reports the results of the proposed method and three other state-of-the-art methods in the field, with the notable exception, neither the training network nor the training dataset are available. As it can be seen, the proposed method has comparable performance for motions less than 5 pixels - for larger motions as it is largely expected from methods that rely on coarse-to-fine schemes that involve downsampling it has lower accuracy.

References

[1] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004. 1

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 1

[3] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2014. 1