Learning Optical Flow For Action Classification

Çağatay Odabaşı[1]

***Abstract*— Action recognition is the task of assigning labels to human actions. It is particularly important for service robots because they need to react to human actions. For instance, if the user is cooking, the robot can offer to bring him some tools such as a pan or knife. In this work, the possibility of learning action recognition and optical flow extraction simultaneously using 3D Convolutional Neural Networks is analyzed. The preliminary results show that it is possible to learn two tasks together, but the proposed architecture needs further improvements.**

I. Introduction

People are performing a lot of different actions such as repairing something, walking, watching tv, eating, sleeping, taking medications, cooking, taking care of a baby. While performing them, they would need some external help. For instance, in Fig. 1, Lisha is taking care of a baby. So, she may need some help. In these situations, giving commands to robot would be too hard and the person may choose not to ask help from the robot. If the robot can understand these needs just by observing the people, it can offer some help without any explicit command. This would make user's life easier.

The action recognition is a task of labeling the data stream (video, image, skeleton, etc.) with an appropriate label such as walking, watching tv, etc. The main input source is a video stream which contains both spatial and temporal information. That's why the general approach is to exploit both domains to achieve high accuracy. To do this, the optical flow, which is an entity representing the displacement of certain part of the image, should be extracted from the video, because it contains rich temporal information content. The current focus is learning both tasks by using Convolutional Neural Networks (CNN) [11].

In machine learning, the loss function is an entity that assigns some cost to certain conditions. When the optimizer minimizes it, it is expected that the network will behave as requested. For example, if the loss function penalizes the misclassified actions, it is expected that the network will learn how to classify the actions correctly when the associated loss function is minimized.

In this work, the main aim is to investigate in learning optical flow and action recognition tasks simultaneously. This would allow us to train the action recognition network on smaller datasets. To do this, a new loss function including both action recognition and optical flow losses is generated. [20] proposed to use a similar cost function; however, they split the problem into two tasks. Instead of splitting the training procedure, the proposed network in this paper is trained by minimizing one joint loss function. The optical flow image is learned internally. So, the sole output of the network is the action recognition label.

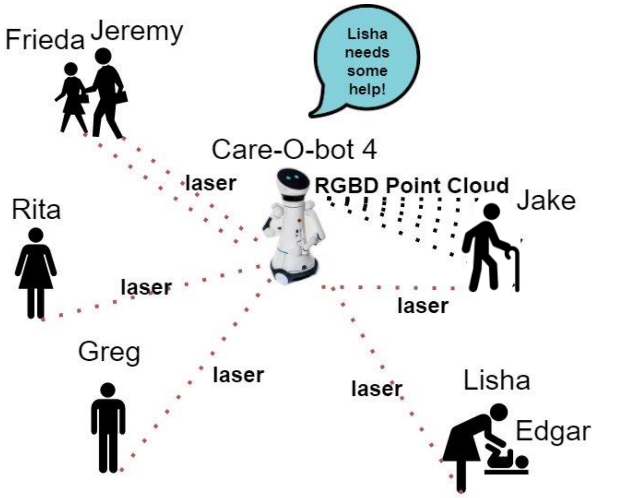

Fig. 1. Service Robots need to observe the people in the environment; so that, they can offer help.

The organization of this paper is as following. First, the related work is introduced to the reader in Section II. Second, the theory behind the proposed network is explained to the reader in Section III. Third, The implementation details are given in Section IV. Also, the preliminary results are presented and discussed in Section IV. Lastly, the paper is briefly concluded in Section V.

II. Related Work

Classical action recognition approaches [19], [18] are tracking temporal trajectories by using optical flow and then they are extracting spatial information such as HOG(Histogram of Oriented Gradient) [2] or spatiotemporal features such as HOF(Histogram of Oriented Flow) [10], MBH (Motion Boundary Histogram) [19] around these temporal trajectories.

The most common CNN architectures for action recognition are two streams networks for RGB and optical flow images [15], [1], [20] and 3D CNN which can convolve the video both in spatial and temporal domains [17], [3], [14]. Even though these approaches outperform on big datasets such as Kinetics [7], Youtube1M [6], they cannot reach the level that hand-crafted features based methods reached on relatively small datasets such as UCF101 [16] and HMDB51 [9]. One possible approach to this problem is fine-tuning. In

[1] The author is with Robot and Assistive Systems Department at Fraunhofer IPA, 70569 Stuttgart, Germany cagatay.odabasi@ipa.fraunhofer.de

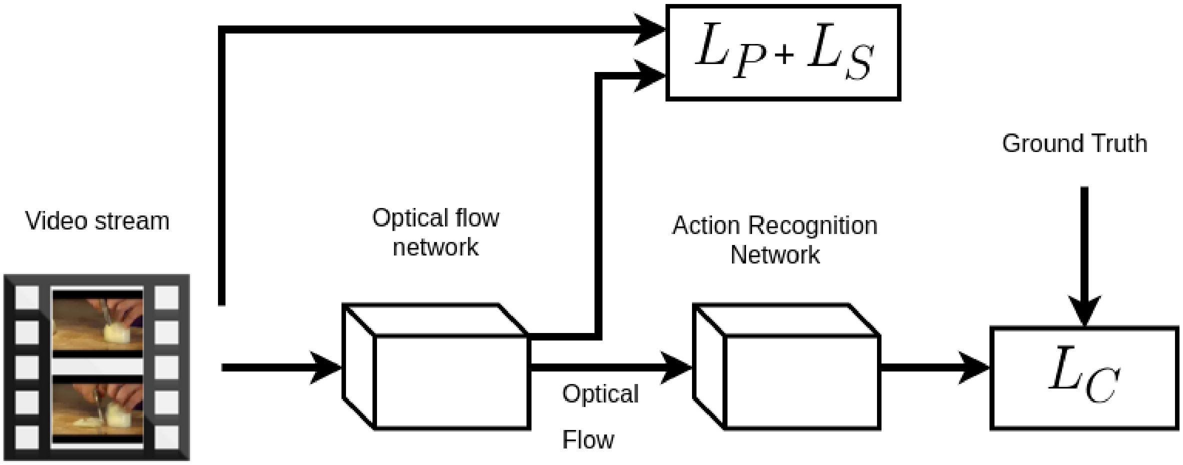

Fig. 2. The optical flow network accepts the video stream as input and its output is optical flow which is directed to the action recognition network. During the training, the optical flow loss and classification loss are combined and minimized together. The ground truth is true action recognition labels provided by dataset.

this approach, first, the network is trained on big datasets, then the classifier part of the network is retrained on a small dataset of interest. This approach would work if the datasets are similar to each other. If the small dataset differs dramatically from the big one, then there is no point of fine-tuning. Also, gathering a high amount of video data would be time-consuming. Another approach would be using optical flow as an input to the network which makes it possible to train on a small dataset [12].

For calculating optical flow efficiently inside the network, it's been proposed to use classical optical flow equations as a loss function [5], [20]; so that, the network can learn how to extract optical flow just by using input frames along with its own output without any supervision. They are basically warping the second frame by using the estimated optical flow and a differentiable warping function. Then, this warped image is extracted from the first image to calculate the photometric error.

In this work, the structure proposed in [20] is used; however, this work differs from them by trying to learn optical flow and action recognition simultaneously rather than two different tasks and also 3D convolutions are used rather than 2D, since 3D convolutions are capable of exploiting the spatial and temporal domains simultaneously by convolving them in both domains.

III. Methodology

A. Estimating Optical Flow

The part of the network that estimates the optical flow is called optical flow network. The rest is called action recognition network. The optical flow network is a 3D convolution based encoder-decoder network which is placed in front of the action recognition network as in Fig.2. The optical flow network takes three consecutive frames at a time and produces one optical flow for each set of input. Let's call these grayscale input frame at time step k as I_k. Clearly, the optical flow network is fed by I_{k+1}, I_k and I_{k-1}.

The loss function should be adjusted so that when it is minimized, the network should learn both optical flow and classification loss. Classification loss is the classical cross entropy loss function so let's refer to it as $\mathcal{L}_{\mathcal{C}}$. It penalizes misclassified actions.

Our optical flow loss function consists of photometric loss and smoothness loss. Smoothness loss is required for the aperture problem, so smoothness loss will force the system to learn just small motions.

The photometric loss is defined as:

$$\mathcal{L}_{\mathcal{P}} = \rho(W(I_k, O_k) - I_{k-1}) \tag{1}$$

where $\rho = (x^2 + \epsilon^2)^{1/p}$ is the Charbonnier cost, $W(I_k, O_k)$ is the warping function which warps the input image I_k by using optical flow O_k, so that, it will be identical to I_{k-1}. The implementation of this function is adapted from grid sampler of [4]. The warping function is sampling one pixel for each pixel position in the new image from the input image by using a flow field. This flow field indicates the displacement of each pixel of the input image.

The smoothness loss is defined as:

$$\mathcal{L}_{\mathcal{S}} = \rho(\nabla_x O_{x,k}) + \rho(\nabla_y O_{x,k}) + \rho(\nabla_x O_{y,k}) + \rho(\nabla_y O_{y,k}) \tag{2}$$

where ∇_x, ∇_y are the horizontal and vertical gradient operators applied to horizontal$O_{x,k}$ and vertical$O_{y,k}$ components of optical flow.

The general loss function can be written as:

$$\mathcal{L}_{\mathcal{G}} = \alpha_1 \mathcal{L}_{\mathcal{C}} + \alpha_2 \mathcal{L}_{\mathcal{P}} + \alpha_3 \mathcal{L}_{\mathcal{S}} \tag{3}$$

where $\alpha_{1,2,3}$ are manually selected constants which arranges the magnitudes of different losses. Therefore, the optimizer minimizes the joint loss $\mathcal{L}_{\mathcal{G}}$.

network can fit the training set completely, but it cannot get good results on validation dataset. To avoid this, we introduce Color Jitter on training videos. After each frame is normalized which is a general method for most of the CNNs, the Gaussian noise is added to them as below:

B. *Stacking the optical flow images*

The proposed architecture is presented in Fig.2. As seen, the only modality that action recognition network uses is the output of the optical flow network. Since the action information is spread through the entire or a part of the video, it is not possible to make a good prediction with just one optical flow. Therefore, the optical flow outputs must be stacked.

In both training and testing mode, the network accepts a fixed number of sequential frames. These frames are divided into smaller groups where each group contains three frames. Each of these groups is sent to the optical flow network and as a result, it produces one optical flow image. These optical flow images are stacked in the time axis and are sent to the action recognition network to get the action recognition output.

IV. Results

A. *Implementation Details*

For the training part, 2 Nvidia GTX1080Ti with a batch size of 256 are used. Also, the image size is reduced to 56x56 to make the system memory efficient. The action recognition is done by using 3D-Resnet-18 proposed in [3]. A small 3D CNN network is added in front of it to infer the optical flow. It is kept as simple as possible due to computation power. As optimizer, we use Adam [8] presented in PyTorch framework [13], since Adam is easier to tune than Stochastic Gradient Descent (SGD) and its performance is comparable to SGD.

B. *Evaluation*

In this section, the preliminary results are presented. Therefore, a detailed evaluation should be carried out to assess the optical flow and action recognition performances. The aim is to maximize the classification accuracy of the system while minimizing the optical flow loss. That's why the loss and accuracy results of the model are presented. Note that the results are preliminary. The architecture is still in development. Current action recognition scores cannot reach the state-of-the-art level which is around 45% without fine-tuning. However, they prove that it is possible to learn two tasks simultaneously.

In Fig. 3, the training accuracy, training loss, validation accuracy, and validation loss are presented. The final values of the loss function on both sets are around 40. This means that the network can generalize well. However, the validation accuracy can reach up to 27%, although the training accuracy is around 90%. Therefore, we can conclude that the problem is in learning action recognition rather than optical flow. There could be several reasons for this. The most important one would be the stacking the optical flow images. The number of frames stacked would not be enough. On the other side, if the number of frames is increased, the computation cost and memory consumption increase dramatically.

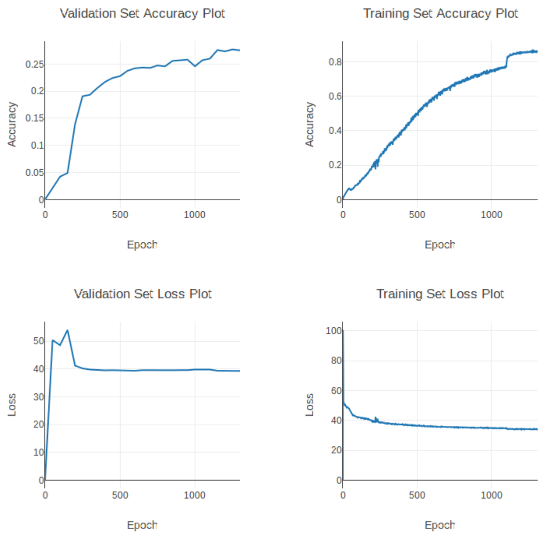

Fig. 3. The accuracy and loss results of action recognition network with optical flow part. The accuracy is normalized to [0,1]. So, the accuracy percentage can be calculated by multiplying the values with 100.

V. Conclusion

In this work, the possibility of learning unsupervised optical flow and action recognition tasks simultaneously is tested. To do this, a joint loss function is created which consists of both optical flow loss and the action recognition loss. Minimizing the overall loss function would allow the network to learn two tasks simultaneously.

The proposed method would allow us to learn with smaller datasets and our results show that the joint loss function can be minimized simultaneously. However, the action recognition performance is still too low, hence it needs some further analysis.

In future works, the loss function will be analyzed in detail. The findings will help us to optimize the architecture.

VI. Acknowledgement

This work has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 721619 for the SOCRATES project. Also, I want to thank all of my colleagues in the Domestic and Personal Robotics Group at Fraunhofer IPA for their ideas and support.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

[2] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[3] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *Proceedings of the ICCV Workshop on Action, Gesture, and Emotion Recognition*, volume 2, page 4, 2017.

[4] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[5] J Yu Jason, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016.

[6] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[7] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Hilde Kuehne, Hueihan Jhuang, Rainer Stiefelhagen, and Thomas Serre. Hmdb51: A large video database for human motion recognition. In *High Performance Computing in Science and Engineering 12*, pages 571–582. Springer, 2013.

[10] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[11] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, pages 319–345. Springer, 1999.

[12] Joe Yue-Hei Ng, Jonghyun Choi, Jan Neumann, and Larry S Davis. Actionflownet: Learning motion representation for action recognition. *arXiv preprint arXiv:1612.03052*, 2016.

[13] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[14] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5534–5542. IEEE, 2017.

[15] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[16] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[17] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 4489–4497. IEEE, 2015.

[18] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.

[19] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, Sydney, Australia, 2013.

[20] Yi Zhu, Zhenzhong Lan, Shawn Newsam, and Alexander G Hauptmann. Hidden two-stream convolutional networks for action recognition. *arXiv preprint arXiv:1704.00389*, 2017.