Attention-based performance improvement for gaze localization

Axel Straminsky, Daniel Acevedo, María Elena Buemi, Julio Jacobo-Berlles
Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales, Departamento de Computación.
CONICET-UBA. Instituto de Investigación en Cs. de la Computación (ICC).
Ciudad Autónoma de Buenos Aires, Argentina.
mebuemi@dc.uba.ar

Abstract

The recognition of activities and their localization in videos captured with cameras mounted on a person constitutes an area of importance since it has multiple applications: health care, non-invasive monitoring, vision in robots, among others. The recognition of actions captured by these devices requires recognition of objects and their locations. In general, this problem is dealt with using powerful equipment such as clusters or several GPU units. The objective of this work is to propose additions to Lu's algorithm in order to obtain the comparable results with smaller resources. To this end, we have adapted an Action Recognition model in egocentric videos based on Attention Mechanisms combined with Optical Flow. This is an architecture model that uses 2 sub-models in parallel: one based on Optical Flow and the other based on the video itself (RGB images), to which the following improvements were introduced: addition of mixed precision in the training cycle, use of Ranger Optimizer instead of vanilla SGD in the attention mechanism, and the use several activation functions (Swish [10], GELU [3], LeakyReLU and Mish) . For the tests, the EGTEA Gaze+ dataset was used, which consists of videos of first-person actions from daily life and the experimentation carried out together with the results achieved. This leaves open the possibility of testing more complex datasets.

1. Introduction

The increasing use of wearable devices such as Go-pro, HoloLens, contributes to the existence of new datasets that are added to those provided by robotic vision and augmented reality. These videos capture the vision data from the point of view of a human being where gaze changes are made with non-controlled speed. Understanding the scene in these videos and recognizing the action that takes place in it using third-person videos is different from doing the same thing using cameras that observe the scenes from another point of view. First person point of view allows access to the activities performed by a user carrying a wearable device. In recent years, this type of capture has spread trying to understand the environment and to predict activities by analyzing fine motor skills, such as hand-object manipulation or eye-hand coordination [1]. Thus, attention is focused on the space-time location where an action takes place, through gaze tracking and hand-object location algorithms. Given the increase in the number of egocentric videos every day, more challenges arise, such as activity recognition/classification, video summarisation and object detection [9].

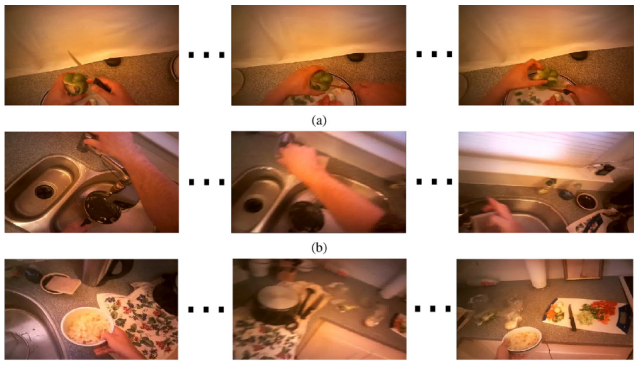

Figure 1. Examples of egocentric actions (subsampled frames) from the Extended GTEA Gaze + dataset: (a) 'cut bell pepper' action, (b) 'wash pan' action and (c) 'move bowl' action [9].

This paper deals with two main aspects of scene understanding: gaze localization and hand/object detection.

With regard to gaze localization we focus on the work done by Lu et al. [6] that uses a model with an architecture based on 2 sub-models (also called flows) in parallel, one of them to recognize movement (using Optical Flow) and the other to recognize the objects in the image. The authors have introduced a spatio-temporal attention mechanism called STAM (Spatio-Temporal Attention Module) in each of these flows, and a training process that uses gaze supervision (that is, guided by the gaze of the camera).

This turns out in a better recognition performance since the model concentrates more on the object of interest, leaving aside the background noise. In order to introduce ourselves to the challenge of obtaining recognition rates close to those of Lu [6] we performed several experiments in which we significantly reduced training time and the number of iterations. Taking as reference the results obtained for RGB and Optical Flow, this results open a possibility to advance in this methodology despite of the limited resources available. To improve training time, we used Mixed Precision, which allows training with a batch size larger than the one originally used in [6], thus accelerating convergence, with a very small loss in precision. On the other hand, for precision, we tried different variants of the activation functions, and also with changes in the number of layers. In addition to this, we tested different optimizers that are more complex than the one used in the original implementation (vanilla SGD), with the aim of improving both convergence time and precision in the evaluation stage.

2. Methodology

The methodology proposed in this section aims to improve both performance in training time and the accuracy of the model by [6]. We used Mixed Precision for performance improvement, making it possible to train with larger batch sizes than the one originally used in [6]. In this way it is possible to speed up the convergence with a small loss in precision. On the other hand, in terms of precision, we show different variants of the activation functions, and also changes in the number of layers of the attention mechanism. We also present different optimizers that are more complex than the ones in the original implementation (vanilla SGD), with the aim of improving both the convergence time and the accuracy in the evaluation stage.

2.1. Mixed Precision

This technique [7, 8] uses both 32-bit and 16-bit float data during training, with the aim of speeding up training while consuming less GPU memory, and experimentally, no losses in precision are usually observed.

In this algorithm the first step consists of casting the weights of the network from FP32 to FP16, and doing the forward and backward pass using FP16. However, due to this, the gradient can end up being very small, so when updating the weights themselves, this is done in FP32, previously scaling the error (loss) by a factor.

2.2. Optimizers

The default optimizer (SGD) is replaced with a more advanced one called Ranger, which is a combination of the LookAhead method [11] and the Rectified Adam (RAdam) method [5]. RAdam is a variant of the Adam optimizer, which adds a dynamic adjustment of the learning rates and in this way it reduces the variance of the training weights, which improves the convergence of the model towards a better local minimum than would be achieved with the standard version of Adam.

LookAhead is a method inspired by recent advances in the understanding of error surfaces, and consists of maintaining two sets of weights and interpolating between them, allowing one of these sets to update faster, using it for exploration, while using the other set of slower updating weights for long-term stability. The result of this is a reduction in variance and less sensitivity to a suboptimal choice of hyperparameters during training, as well as an acceleration in convergence.

2.3. Activation Functions

Another of our proposed enhancements include changing the default activation functions (ReLUs) with different types and more complex activation forms. The activation functions we experimented with were: Swish [10], GELU [3], LeakyReLU and Mish.

3. Experiments and Results

The Georgia Tech Self-Centered Activities (GTEA) dataset contains seven types of daily activities, such as making sandwiches, preparing tea or coffee. Each activity is carried out by four different people [2, 4], for a total of 28 videos. For each video, there are about 20 instances of detailed actions, like taking a bread or pouring ketchup. These videos have a duration of approximately 1 minute.

The original paper from Lu [6] trains the model in 64000 iterations. This would imply a very long training time with our current hardware. With our improvement, we retrained and detected that 600 iterations were enough and its performance was evaluated again with that number. In this case and in all subsequent experiments, the models are trained with mixed precision, since this allows a considerable acceleration of the training time required to reach an acceptable solution. In all cases, the original learning rate reduction strategy, MultiStepLR, which reduced the learning rate only in iterations 300 and 1000, was replaced by ReduceLROnPlateau, which reduces it by a factor of 0.5 each time the model does not improve its performance in the validation set for 5 iterations in a row.

The results with the test set from EGTEA dataset are shown in Table 1 for RGB input and Table 2 for Optical Flow. Second row shows our experiments with the same methodology as in [6] but with 600 training iterations. On row three, we have changed the SGD optimizer used by Lu for the Ranger optimizer. On the next experiment we have the expanded Attention Mechanism (with Ranger Optimizer). Finally, when trying different activation functions, we report the best results which were achieved with the Leaky ReLU.

	Accuracy	Mean class Acc.
Lu [6]	0.6356	0.5634
Lu (600 iterations)	0.5479	0.4386
Ranger Optimizer	0.6102	0.5456
Expanded Att. Mechanism	0.5989	0.5323
Leaky ReLU Act. Fc.	0.6122	0.5478

Table 1. Accuracy and Mean Class Accuracy on RGB test set from EGTEA dataset. From row 2 to the bottom, 600 iterations were performed on all experiments.

	Accuracy	Mean class Acc.
Lu [6]	0.6009	0.5099
Lu (600 iterations)	0.4881	0.3461
Ranger Optimizer	0.5588	0.4544
Expanded Att. Mechanism	0.5618	0.4617
Leaky ReLU Act. Fc.	0.5657	0.4592

Table 2. Accuracy and Mean Class Accuracy on Optical Flow test set from EGTEA dataset. From row 2 to the bottom, 600 iterations were performed on all experiments.

4. Conclusions and future work

In this work we proposed additions to Lu's algorithm in order to obtain the comparable results with smaller resources. This approach uses Lu's double flow methodology based on the attention mechanism; each flow was studied separately aiming at obtaining similar results with much less computational complexity. As it can be seen in both tables of this paper, the accuracies obtained for each of the proposed improvements are comparable to the ones corresponding to the original Lu's [6] paper.

References

[1] Alireza Fathi, Yin Li, and James M. Rehg. Learning to recognize daily actions using gaze. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 314–327, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[2] Alireza Fathi, Xiaofeng Ren, and James M. Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288, 2011.

[3] Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs). 2016.

[4] Yin Li, Zhefan Ye, and James M. Rehg. Delving into egocentric actions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 287–295, 2015.

[5] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond, 2020.

[6] Minlong Lu, Danping Liao, and Ze-Nian Li. Learning spatiotemporal attention for egocentric action recognition. Oct 2019.

[7] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. *CoRR*, abs/1710.03740, 2017.

[8] Nvidia. Mixed-precision training of deep neural networks. Oct 2017.

[9] Adrián Núñez-Marcos, Gorka Azkune, and Ignacio Arganda-Carreras. Egocentric vision-based action recognition: A survey. *Neurocomputing*, 472:175–197, 2022.

[10] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2018.

[11] Michael R. Zhang, James Lucas, Geoffrey Hinton, and Jimmy Ba. Lookahead optimizer: k steps forward, 1 step back, 2019.