# First Person Activity Recognition exploiting a Self-Supervised motion task, frame selection, a double Conv-LSTM and LSTA

Emanuele Fasce
Politecnico di Torino
S277983
emanuele.fasce@studenti.polito.it

## Abstract

*In this paper several neural architectures for first person action recognition are implemented and evaluated on the GTEA61 dataset. First, the Ego-RNN network is implemented and a joint training exploiting a spatial attention mechanism using a conv-LSTM as well as the warped optical flow. Secondly, a self supervised task that exploits IDT maps is then added to the appearance stream of the Ego-RNN network. Lastly the LSTA network is implemented. The proposed original variations are a frame selection preprocessing procedure and the Double Conv-LSTM.*

## 1. Introduction

The automated classification of an action from a first-person video has become a very important task in computer vision also due to the recent diffusion of first-person cameras and wearable devices among the general population. An accurate and efficient algorithm could indeed impact several applications, like robotics, indexing and retrieval, workers security, human computer interaction, to name a few. However, first-person video action recognition not only entails difficulties due the video recognition task, like the huge computational cost and the need for capturing spatial-temporal context between frames, but also specific challenges related to the first-person video, like the camera motion due to the subject movements which are not necessarily correlated with the performed actions. All these aspects had led to the development of specific network architectures, some of which are going to be discussed in this work. It is also worthwhile to underline that usually networks have been trained to recognize generalized motion (put, take), but the dataset used, GTEA61, contains videos of very fine-grained activities like putting coffee or taking coffee, making this task even more challenging due to the need of a network capable of understanding the order among the frames and of modeling the relationship between the hands movements and the involved objects.

### 1.1. Related works

Deep neural networks were not extensively used in video recognition tasks until in 2014 two breakthrough papers introduced two different approaches whose structure is still considered today: single stream and two streams networks. In the Ego-RNN network paper [3] proposed in 2018 by Sudhakaran et Al. the researchers followed the latter approach by using an appearance stream encoding RGB frames, augmented with an attention mechanism, and a temporal stream encoding stacked optical flow. The same researchers the year after updated their two stream networks with a variation of the convLSTM called LSTA [2] which introduced attention pooling and output pooling. The disadvantages of this architecture is that the two streams are trained separately and jointed together with a fully connected layer. To overcome this limitation, Planamente et Al. proposed a single training network that adds a self supervised task exploiting (IDTs) to the RGB stream backbone [1]. This work is focused on the implementation and the evaluation of these networks on the GTEA-61 Dataset and some original variations called frame selection and Double Conv-LSTM. The code is available at https://github.com/emanueleing/FPAR

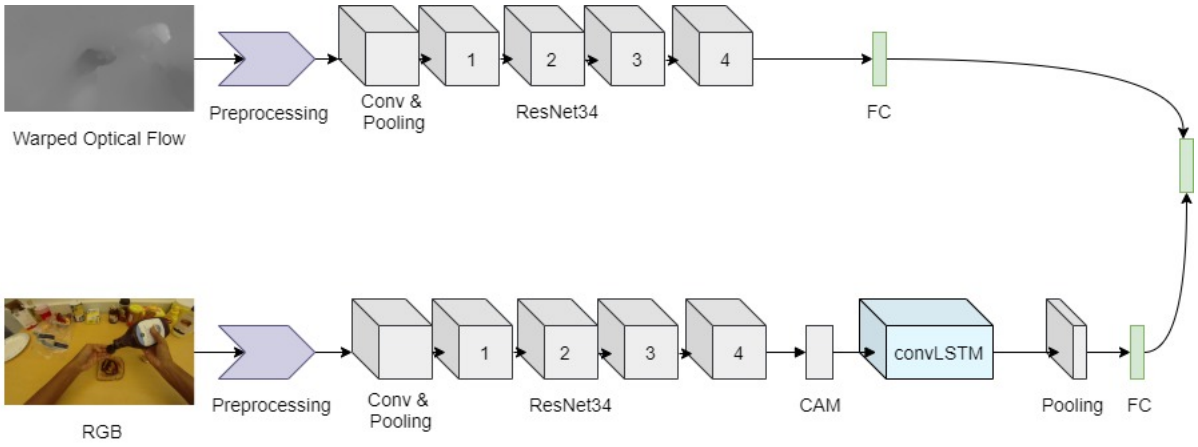


Figure 1. Ego-RNN architecture.

## 2. Architecture overview

### 2.1. Ego-RNN

Ego-RNN is a CNN-RNN architecture which has been proposed by Sudhakaran et Al. whose backbone is composed of a Resnet34 pretrained for generic image recognition and augmented on top with an attention mechanism that uses class activation maps for spatially selective feature extraction. The memory tensor of a convolutional LSTM tracks the discriminative frame-based features distilled from the video for activity classification. The usage of a conv-LSTM, a modified version of LSTM where matrix products have been substituted by convolutions in gates, helps to maintain the spatio-temporal encoding of objects and their locations into the descriptor. The researchers also trained a temporal network that uses stacked warped optical flow (optical flow that has been corrected for the movement of the subject holding the camera) images as input, in order to encode the motion changes occuring in the videos. These two networks are trained separately and then after the merge due to a fully connected layer trained jointly. In Figure 1 a graphical representation of the entire architecture is shown.

### 2.2. Self-supervised motion segmentation task

Planamente et Al. proposed a single stream architecture called Self-supervised first Person Action Recognition network (Sparnet) which adds a motion segmentation (MS) self-supervised task to the Ego-RNN architecture. In the MS task the discrepancies between a binary map labeling pixels as either moving or static and the object movement predicted by the network when it looks at a single frame are minimized, forcing the CNN to learn an image embedding that focuses on object movements. This self-supervised task is applied on the second stage of the RGB stream of the Ego-RNN network as indicated in Figure 2. The features extracted from the fourth layer of the Resnet are sent to a convolutional layer, called MS Head, which reduces the 512 channels into 100 channels, followed by a fully connected layer and a softmax. The per-pixel binary Cross-entropy loss (Lms in the picture) is computed between the output and the down-sampled ground-truth labels. These labels, which are obtained by extracting the motion information exploiting the Improved Dense Trajectories (IDT) maps, in the preprocessing phase are down-sampled to 7x7 using the Resize transform. The final loss is the sum of the per-pixel binary cross-entropy loss Lms, weighted by a parameter alpha, and the standard cross-entropy loss Lc computed between video predictions and video labels. Since information could be lost by using a threshold on the IDT maps, a regression variation is built by adding a sigmoid at the end of the end of the MS task and substituting the binary cross entropy with a regression loss function. Several loss functions with different hyper-parameters combinations are tried in next sections.

### 2.3. Frame selection

An implemented variation is based on the assumption that frames that contain more movement are more useful to represent the activity than the others. In this case IDT maps are exploited for a very simple preprocessing task called frame selection. Instead of sampling K frames uniformly from videos, the K frames with the brightest maps are selected for each video. This procedure could be helpful especially in scenarios where for each video there are a lot of frames available and the relevant action is compressed in a few of them or when hardware constraints limit the number of processed frames. Even if this preprocessing procedure is simple and flexible, it requires the computation of IDT maps and the effectiveness of it could depend also on the type of analyzed video, since it is not always the case that

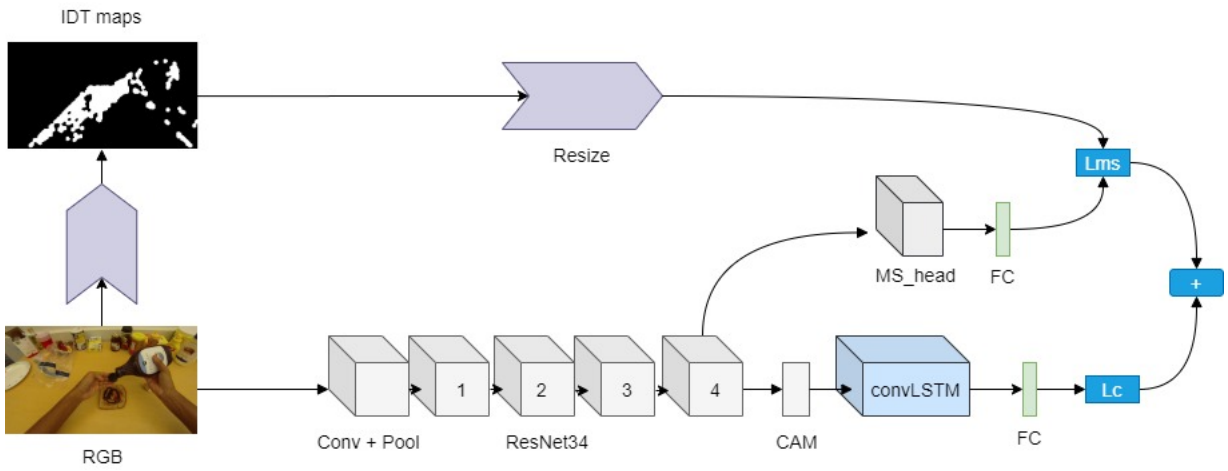


Figure 2. Self-Supervised Motion Segmentation task architecture

the frames with more movement inside are more useful.

### 2.4. Double Conv-LSTM

This variation is based on the idea that egocentric and especially manipulation activities involve the interaction of more than one object present in the video. For example, in GTEA-61 one hand or also two hands are used to perform an action on a object, like opening or taking it. However, the input of the LSTM is only the CAM of the winner class, so the LSTM could not be able to track the positions of more objects at the same time, leading to a possible loss of information. In this simple approach the conv-LSTM is replaced with two Conv-LSTM merged through a fully connected layer. Obviously, the choice of using two conv-LSTM is given by the hardware constraints, but additional experiments could be performed using more networks or more input combinations. The inputs used for these experiments are the CAMs of the first together with the second winner class given by the Resnet, and the CAM of the first winner class together with the raw output of the 4th convolutional layer. The first choice is justified by the need to track more and possibly different objects at the same time, while the second combination of inputs is inspired by the LSTA architecture (described in the next section) which performs a Hadamart product between the CAM and the normalized output of the convolutional layer. The main drawback of this architecture is the increased number of parameters to train.

### 2.5. LSTA

Long-Short Term Attention is a network proposed by Sudhakaran et Al. a few years after the publication of their Ego-RNN model. Two important elements characterize this network are a built-in attention mechanism and an output pooling mechanism. The attention pooling enables the network to identify the relevant regions in the input frame and to maintain a history of the relevant regions seen in the past frames, while the output pooling enables the network to propagate a filtered version of the memory which is localized on the most discriminating components. One of the innovations of this architecture consists in moving the attention mechanism from the CNN level to the RNN.

## 3. Implementation details and results

For what concerns the Ego-RNN architecture, the temporal stream is trained in only one stage. The total number of epochs is 700, the learning rate is 1e-2, decayed by 0.5 after 150, 300 and 500 epochs. The RGB stream instead is trained in two stages. In the first stage, only the conv-LSTM and the classifier are trained. Among the chosen hyper-parameters, the number of total epochs is 200, the initial learning rate is 1e-3 and is decayed of 0.1 after 25,75 and 150 epochs. In the second stage also the fourth convolutional layer of the ResNet is trained. The total number of epochs is 150, the learning rate is 1e-4 and is decayed of 0.1 after 25 and 75 epochs. In the joint training model the two streams are merged with an extra fully connected layer, and training lasts for for 100 epochs using 1e-4 as the learning rate. Table 1 reports the results obtained with these models by uniformly sampling 7 and 16 frames from the videos. Hardware constraints limit the possibility of using 25 frames as in the original paper by Sudakaran et Al. Even though the accuracy is slightly lower, the results reflect the findings of their paper, since there is a significant increase in accuracy when exploiting the attention mechanism with respect to using just the raw output of the convolutional layer of the ResNet, and when merging the two streams in the joint training model. The best accuracy reached when merging the two streams is indeed 71.55%.

| Accuracy | 7 frames | 16 frames |
|---|---|---|
| ConvLSTM | 37.93 | 54.31 |
| ConvLSTM + attention | 53.44 | 63.79 |
| Warped Optical Flow | 38.79 | 42.24 |
| Joint train | 64.65 | 71.55 |

Table 1. Ego-RNN results.

In the Self-Supervised motion task, an extensive grid-search (reported in Table 2) is performed in order to find the value for the kernel size and padding and the loss that work better on the validation set. The combination of hyper-parameters that works better for regression is used to perform a second grid-search that looks for the best value of alpha.

$$Loss = L_{classifier} + \alpha L_{MotionTask}$$

The hyperparameters used are the same as the ones used in the second stage of the RGB stream in the Ego-RNN architecture. Among the regression losses used, apart from the typical MSE and MAE losses, there is the Kullback-Leibler divergence.

| Loss | Kernel size | Padding | Accuracy |
|---|---|---|---|
| Cross-entropy | 1 | 0 | 66.37 |
| Cross-entropy | 3 | 1 | 68.96 |
| Cross-entropy | 5 | 2 | 64.65 |
| MSE | 1 | 0 | 62.93 |
| MSE | 3 | 1 | 63.97 |
| MSE | 5 | 2 | 67.24 |
| MAE | 1 | 0 | 61.20 |
| MAE | 3 | 1 | 62.93 |
| MAE | 5 | 2 | 62.93 |
| KLDiv | 1 | 0 | 63.79 |
| KLDiv | 3 | 1 | 62.93 |
| KLDiv | 5 | 2 | 62.06 |

Table 2. Self supervised motion task results, loss and kernel size search

| Loss | Kernel size | Alpha | Accuracy |
|---|---|---|---|
| MSE | 5 | 1 | 67.24 |
| MSE | 5 | 0.1 | 64.65 |
| MSE | 5 | 0.01 | 65.51 |
| MSE | 5 | 0.001 | 62.06 |

Table 3. Self supervised motion task results, alpha search

$$Kullback\ Leibler\ D(P|Q) = \sum_{x \subset X} P(x) log\left(\frac{P(x)}{Q(x)}\right)$$

In the frame selection procedure, for each RGB frame the associated IDT map is converted as a black and white tensor and a decreasing ranking of the sums of the resulting tensors is built, so that the brightest IDT map comes first. The RGB frames corresponding to the first K IDT maps are then taken following their original order. The accuracy with K=7 is 62.93 with K=16 is 66.37%. In Figure 3, an intuition about how this procedure helps to select the most useful frames is provided by showing the selected frames with uniform sampling and with frame selection (K=7). The double conv-LSTM architecture consists in two conv-LSTM identical to the original one used in Ego-RNN, whose outputs are sent to different average pooling layers, concatenated, and then sent to a fully connected layer for the prediction. The training phase is composed by the same two stages found in Ego-RNN and the hyper-parameters used are the same as the ones used for Ego-RNN. The accuracies obtained are 63.79% and 68.96% respectively with 7 and 16 frames for video. Also the LSTA network is trained with the same hyper-parameters used in Ego-RNN following its two stage training procedure. A more accurate description of

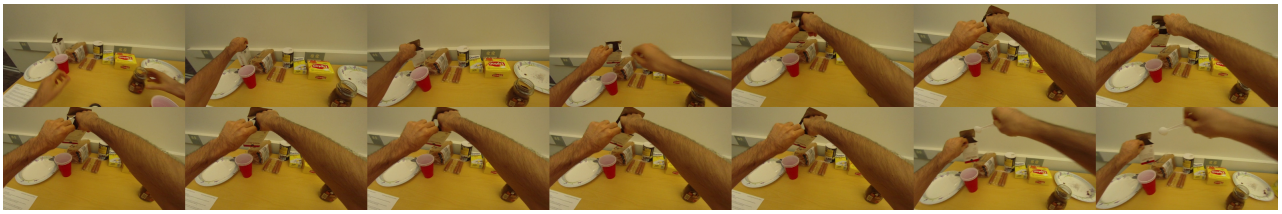

Figure 3. On the top, uniform sampling for take spoon, on the bottom, frame selection

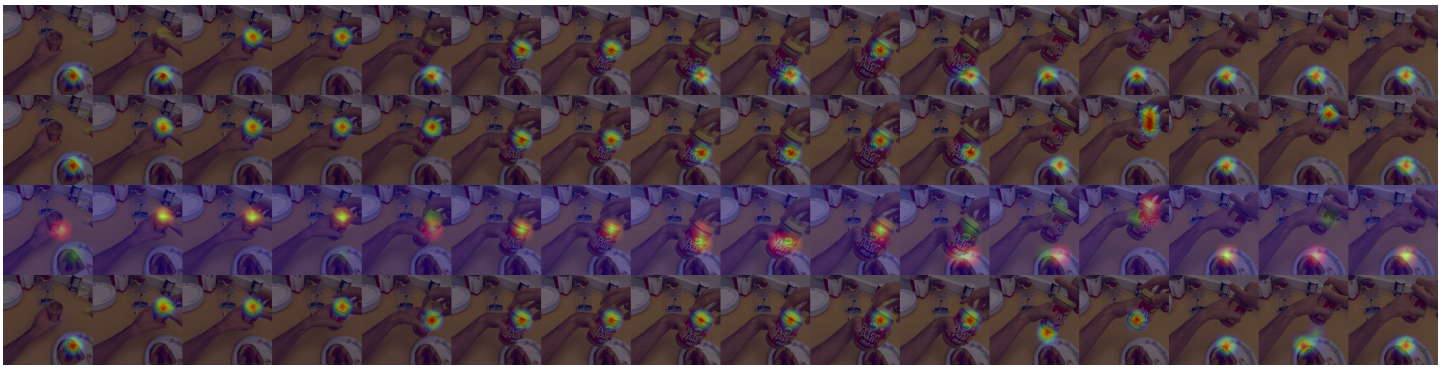

Figure 4. From top to bottom: CAM of Ego-RNN, Self-Supervised Motion Task, Double Conv-LSTM, temporal CAM of LSTA

the LSTA architecture is now provided following the code of the authors. In this network, The CAM derived from the fourth convolutional layer or the Resnet is sent to a convolutional RNN, whose output is summed to the CAM itself. A Hadamart product is taken between this sum, on which a softmax has been applied, and the feature map X coming from the fourth convolutional layer of the Resnet. This procedure is the built-in attention mechanism. The conv-LSTM takes as input the Hadamart product, which is also sent to a special pooling operation composed of an average pooling and a fully connected layer (called "copling FC" in the code), and its memory state is sent to another average pooling and a fully connected layer (called "c_classifier" in the code). The outputs of the two fully connected layers are summed up and a CAM derived from the c_classifier branch is built. This is multiplied by the memory state of the conv-LSTM and the result is used to update the hidden state. The accuracy found with LSTA using 16 frames is 65.51%. In Table 4 a brief summary of the results is presented.

| Network | Used frames | Accuracy |
|---|---|---|
| Ego-RNN RGB stream | 7 | 53.44 |
| Ego-RNN RGB stream | 16 | 63.79 |
| Ego-RNN RGB stream + MS task | 16 | 68.96 |
| Ego-RNN RGB stream + frame selection | 7 | 62.06 |
| Ego-RNN RGB stream + frame selection | 16 | 66.37 |
| Double Conv-LSTM (2 CAMs) | 16 | 68.96 |
| Double Conv-LSTM (1 CAM + X) | 16 | 67.24 |
| LSTA | 16 | 65.51 |

Table 4. Final results.

### 3.1. CAM visualizations

As explained before, The EgoRNN architecture exploits an attention mechanism by using the Class Activation Maps (CAMs) of each frame. They give an indication of the zones of the picture on which the CNN focuses on when predicting the winning class and can be easily visualized, therefore they can be used to estimate the quality of different architectures. If the CAM visualizations are superimposed to the frames, it can be seen where the CNN focuses in each architecture (Figure 4). Obviously analyzing only one video (in this case the action of taking the peanut butter is visualized) is not enough to compare the different architectures, but some interesting patterns can be seen. Indeed, the CNN of the Ego-RNN focuses on the peanut butter in 8/16 frames, the CNN of the architecture implementing the Self-Supervised motion task in 12/16 frames, the CNN of the double Conv-LSTM in 13/16 frames (even though it's an easier task since two CAMs are visualized), the LSTA in 12/16 frames. The visualized CAM of the LSTA is not built at the CNN level but at the output pooling level of the LSTM, so providing also additional temporal information. It is evident from these visualizations that all the implemented variations improve the performances of the Ego-

RNN network.

## 4. Conclusions

All the implemented variations provided improvements to the baseline Ego-RNN architecture. The Self-Supervised MS task and the LSTA network were proven to be successful even though the accuracy of the original papers are not reached due to the limited number of used frames. The original frame selection method and the Double Conv-LSTM that also improved the performances of the baseline could be further studied and modified in future studies.

## References


[1] Mirco Planamente, Andrea Bottino, and Barbara Caputo. *Self-Supervised Joint Encoding of Motion and Appearance for First Person Action Recognition*. 2020. arXiv: 2002.03982 [cs.CV].

[2] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. "LSTA: Long Short-Term Attention for Egocentric Action Recognition". In: *CoRR* abs/1811.10698 (2018). arXiv: 1811.10698. URL: http://arxiv.org/abs/1811.10698.

[3] Swathikiran Sudhakaran and Oswald Lanz. "Attention is All We Need: Nailing Down Object-centric Attention for Egocentric Activity Recognition". In: *CoRR* abs/1807.11794 (2018). arXiv: 1807.11794. URL: http://arxiv.org/abs/1807.11794.