Feedback Approach to Foster Motion Information in FPAR

Susim Mukul Roy
B20AI043

Vikash Yadav
B20AI061

1. Introduction and Problem Statement

First person action recognition (FPAR) task is one of the most challenging in action recognition field. Most of the existing works address this issue with two-stream architectures, where the visual appearance and the motion information of the object of interest, are exploited. In this project, we address this problem by proposing a different approach that tries to use an end-to-end architecture, composed of a single RGB stream influenced by the motion-prediction (MP) self-supervised task [1]. In addition, we include an feedback mechanism to inject the idea of movement directly in the first layers of the RGB stream, in order to guide the model to pay more attention on what really moves, rather than considering the static background. The code can be found at Github.

2. Methodology

2.1. Architecture Overview.

The starting point of our architecture (Figure 3, action recognition block) is the RGB stream defined by [2]. Firstly, we extract uniformly N sparse representative RGB frames, from each input video segment. These frames are used as input to a pretrained CNN backbone in order to extract the embedded appearances from each frame. The spatial attention layer is used to make the network focused on the regions consisting of the objects. The output is then passed to a ConvLSTM network. The last layers are used for the classification (avgpool, dropout(0.70), fully connected). This stream is used to extract important spatial and temporal information from the video. However, the resulting features are still lacking the motion information which is crucial for FPAR task. This issue is solved in the two-stream approaches relying on explicit optical flow data and solve a single multi-task network. During the training, the network has to solve two different tasks concurrently: the action recognition task and a motion-segmentation (MS) auxiliary task. The second one is formalized as a self-supervised problem which takes as input a single RGB frame and tries to identify which parts of the image are going to move.
This identification task is treated as a labelling problem

Figure 1. Regression with ConvLSTM

Figure 2. Our Variation

which minimizes the differences between the motion map, in which pixels are labeled as moving or not, and the object movements predicted by the network for a single static RGB frame. Following, we leverage the Improved Dense Trajectories IDT [3], for extracting "stabilized" motion information. The self-supervised motion prediction task is used by the architecture in order to benefit from both motion and appearance information, using a single RGB stream. The resulting architecture is lighter than the two stream model with less parameters and it is end-to- end trainable.

Our contribution The main idea is that the low-level RGB features are adjusted by two factors β and γ in order to give more importance to the zones that are classified as "in movement". This is performed by means of the feedback branch, which takes, as input, the features of the motion segmentation task, and outputs β and γ. In order to modulate the appearance net- work, we apply a transform function $\theta(.)$, with the learned β and γ, to the RGB features F^{rgb} computed in the first layers of the backbone

$$\theta(F^{rgb}) = \beta \odot F^{rgb} + \gamma \tag{1}$$

$\odot$ is an element-wise multiplication operation and β, γ and F^{rgb} have the same dimensions. The motion information represented by β and γ influences the appearance network by both feature-wise and spatial-wise manipulations. The complete network is shown in Figure 3. We visualize the effectiveness of our contribution by comparing the class ac-

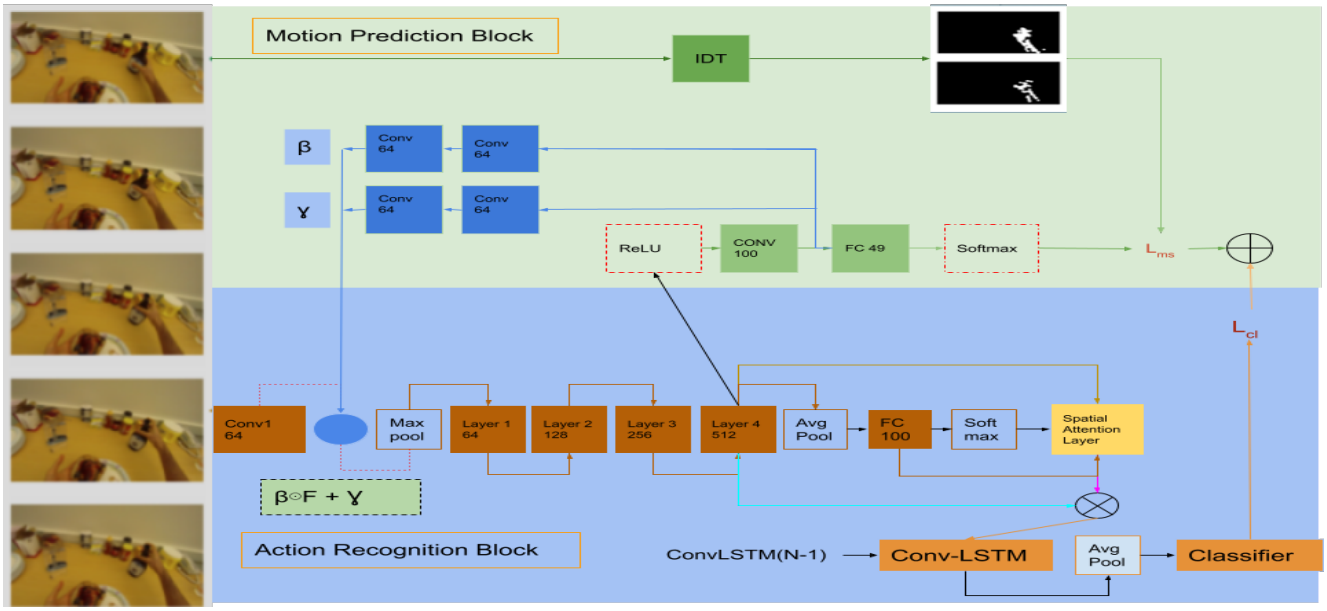

Figure 3. The architecture for the action recogntion task. Here, the Conv 64 layers(in blue) provide the motion information from the later layers into the early layers in order to guide the training as a teacher.

tivation maps(CAM) maps from a ConvLSTM structure before and after applying our variation. We can also visualize the effect of our contribution on the architecture by plotting the class activation maps (CAM) used for the methods used in [2] before and after the insertion of the feedback branch. As we can see from Figure 1, the model predicts some random area of the image too as having action whereas our model,Figure 2 only focusses on the dough part.

2.2. Architecture Details

Mathematical Details. Let T be a training set consisting of $T_i = (H_i, y_i)_{i=1}^n$, where H_i is a set of N timestamped images uniformly sampled from the video segment and y_i is the action performed in that video. Let also be x the output of the model M, depending both on the back bone until and after the Layer-4. Let $g(x)$ be a class probability estimator on the embedding x. We define the categorical cross- entropy classification loss is defined as:

$$L_{cl}(x, y) = -\sum_{i=1}^{N} y_i.log(g(x_i)) \quad (2)$$

The MS branch ends with a fully connected layer of size s^2 followed by a Softmax, and it is trained with a loss L_{ms} based on the per-pixel cross entropy loss between the computed label image l_{ms} and the Ground-Truth motion map m. The estimated l_{ms} is obtained as a function of both image embedding z derived from the backbone until the Layer-4 and MS head parameters. Thus, the L_{ms} loss can be defined as:

$$L_{ms}(z, m) = -\sum_{i=1}^{n}\sum_{k=1}^{N}\sum_{j=1} s^2 m_i^k(j).log(l_{ms,i}^k(j)) \quad (3)$$

The model solves jointly two separate optimization problems: minimize both L_{cl} and L_{ms} (2) (3). L_{cl} affects by back-propagation the Action Recognition head (blocks fol- lowing the Layer4 of the ResNet-34). On the other hand the Motion-Segmentation branch is updated only by its loss. Both losses update the backbone and also the feedback branch of the architecture by a weighted sum of the two through the coefficient α, obtaining :

$$L_{tot} = L_{cl} + \alpha * Lms$$

The value of α is derived by an hyperparameter optimization.

Implementation. The backbone chosen for this architecture is a ResNet-34 pretrained on ImageNet. The output features of the Layer-4 of the backbone, which size is $7 * 7 * 512$, are forwarded also to the Motion Prediction block. The Conv100 block of the Motion Segmentation branch (green in Fig 3.) reduces the feature channels to 100 and, after a flattening operation, the size provided to the fully-connected layer is 49. Meanwhile the output of the Conv100, aforementioned, is also subjected to a Up-Sampling operation. From this we obtain features of size $64*64*100$. This is the input of the two branches by which we calculate β and γ (blue part in Fig 3.). Both of them reduce the channel multiplier to 64 and then are "joined" with the output of the Conv1 with the (1) formula.

3. Experiment

3.1. Dataset

All experiments presented in this project were trained and validated on the GTEA61 dataset that contains a collection of videos, corresponding to 61 actions, performed by 4 dif- ferent users. Inspired by [2], we processed the dataset using denseflow which gives us the warp optical flow in x and y and processed frames of the videos.

3.2. Key Discussions

We present two key factors of implementation in the following points:

- **Prior motion information:** The role of β and γ is to embed motion features into the main stream of the architecture and, to do so, we need to understand at what level is it more convenient to insert this features. We chose at the starting as if we apply the motion features β and γ to the last layers of the ResNet-34, we are going to modify some high level features that are relevant to identify the action and the object involved which won't be helpful.
- **Training structs:** It is necessary to select which blocks to train along with the feedback branch. The intuition is that also the backbone should be partly trained in order to understand this new type of features. A few experiments showed that if we train the entire backbone in the third stage, the outcomes show over-fitting on the training set. Following this, we deduce that the architecture has probably too many parameters to deal with and it is not able to generalize correctly and is subject of further study
- **Streamlit**: We deploy our model on streamlit where the user can input an image or video. In case of an image, we obtain a single class-activation map showing the action region. In case of video, we get a video where the frames are the CAMs overhead the original frames. This can been seen for 4 different models for comparison as per the user's choice.

3.3. Results

Configuration	Accuracy
Temporal Warp Flow	42.24
ConvLSTM	52.56
Two-joint Stream	67.10
Our model	56.48

Table 1. Comparison of results based on accuracies

The proposed variation does not bring the highest accuracy among the state-of-the-art architectures. However, in order to understand if what we did really improved [2], we performed different training with different parts of the architecture. In Table 1 are reported the average results of the obtained accuracy scores. We display the scores obtained by the Ego-RNN's part as well as the validation scores while training the last stage of our variation in Figure 5. All the results are obtained with *number of frames* = 7. Even if our variation does not outperform the Two-joint Stream, our proposed model can be theoretically trained end-to-end, given that we have a lower number of parameters and this brings the model to converge faster as is evident from Figure 4.

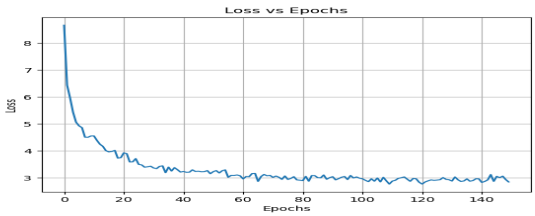

Figure 4. Loss curve

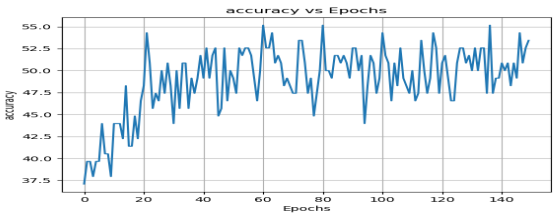

Figure 5. Validation accuracy

4. Conclusion

In this paper, we presented our architecture that is based on the [2] architecture and MS auxiliary task for FPAR. Its improvement is given by the implementation of the feedback branch that helps the architecture to focus more on the motion features, since the beginning.

References

[1] *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*. IEEE, 2021. 1

[2] Mirco Planamente, Andrea Bottino, and Barbara Caputo. Self-supervised joint encoding of motion and appearance for first person action recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8751–8758, 2021. 1, 2, 3

[3] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *2013 IEEE International Conference on Computer Vision*, pages 3551–3558, 2013. 1