International Journal of Multimedia Information Retrieval (2021) 10:219–226
https://doi.org/10.1007/s13735-021-00216-3

REGULAR PAPER

Early-stopped learning for action prediction in videos

Mehrin Saremi[1] · **Farzin Yaghmaee**[1]

Received: 29 January 2021 / Revised: 28 July 2021 / Accepted: 3 August 2021 / Published online: 13 August 2021

Abstract
Action prediction, also called early action recognition, is about recognizing an action in a video with partial observation. Various methods have been developed to tackle either offline or early action recognition, including deep learning approaches. In a family of deep learning methods, video frames or optical flow images are processed sequentially by the network. In this paper, we present a learning framework that can be applied to such methods to make them more appropriate for early recognition. We propose encouraging the learner to learn from earlier parts of the video and stop learning from some point on. By focusing on the earlier parts, we can expect the model to take full advantage of the information lying in these early parts. To this end, it is necessary to find a stopping point up to which enough information has been observed. We measure the amount of information with the help of the loss function. We applied our framework to Temporal Segment Networks and experimented on UCF11 and HMDB51 datasets. The results show that our method improves on Temporal Segment Networks and outperforms other baseline methods.

Keywords Early action recognition · Action prediction · Deep learning · Two-stream networks

1 Introduction

Action recognition is an essential task in video processing that finds applications in detecting crimes, sports video analysis, video retrieval [23], etc. It is essential to recognize an action early in many applications so that necessary actions can be taken.

For example, consider the case of driverless cars. If the autonomous driving system cannot predict what a pedestrian will do next, it will not be able to avoid possible accidents. In surveillance systems, it is crucial to recognize criminal actions early, so that necessary measures can be taken. An elderly monitoring system must raise the alarm in time before something harmful happens to the subject.

In recent years, many techniques have been developed for action recognition. A group of methods are based on the bag-of-words (BOW) technique adapted from the text mining bag-of-words. Such methods extract discriminative key points from the video and describe the neighborhood of these points using descriptors. The descriptors are clustered into prototypes or visual words. Different keypoint detectors have been proposed. Laptev and Lindeberg [17] generalize the Harris corner detector [7] to the spatio-temporal domain, which is sensitive to significant changes in the video. Dollár et al. [5] apply a filter to the video which is sensitive to both dramatic changes and periodic motions. Chakraborty et al. [2] introduce several methods to select a subset of detected interest points.

Bag-of-words methods treat spatial and temporal dimensions similarly while they are of different nature. Trajectory-based methods have been proposed to take more advantage of temporal information [29–31]. In these works, features such as histogram of gradient are extracted in a tube-like area surrounding trajectories. Kantorov and Laptev [10] propose to use motion vector information stored in the compressed video file to estimate optical flow.

As depth sensors become more prevalent, it becomes more feasible to extract 3D positions of human joints and use them to model the skeleton. The moving pose descriptor proposed by [37] is a combination of joint positions (static information) and the first and second derivative of the positions (dynamic information). Qiao et al. [22] introduce "trajectorylet" descriptors, which consist of joint positions,

✉ Farzin Yaghmaee
f_yaghmaee@semnan.ac.ir

Mehrin Saremi
m.saremi@semnan.ac.ir

[1] Electrical and Computer Engineering Department, Semnan University, Semnan, Semnan Province, Islamic Republic of Iran

displacements, and speed in a longer range than [37]. Such methods are limited to situations where human body parts can be distinguished clearly.

Works mentioned so far take advantage of handcrafted features. The success of such methods is challenged by camera movements, complex scenes, and limitations of human detection and pose estimation techniques [38]. Deep learning approaches, on the other hand, are based on automatically extracted features. Among deep learning methods, two-stream networks have been very successful. This kind of network, first proposed in [26], is composed of two convolutional networks, for two modalities, i.e. RGB (spatial) and optical flow (temporal). Frames are fed to the network in sequence, and the results are combined. Wang et al. [33,34] propose a kind of such network called Temporal Segment Networks. In their works, instead of running the model on the whole video, only a sparse sample of short snippets is presented to the network.

In the action prediction literature, [25] proposes two variants of bag-of-words methods, namely integral BOW (IBOW) and dynamic BOW (DBOW). Another BOW-based technique has been presented in [1]. Some of the methods are based on a global-local model [11,12,16]. The local component models an individual segment, whereas the global part models several segments from the beginning. Deep architectures have also been of interest for prediction. Reference [14] proposes a generative adversarial network (GAN), [20] introduces a long short-term memory network (LSTM), and [9] uses a recurrent neural network (RNN) for action prediction.

In this work, we propose a new learning framework that can be applied to many deep learning algorithms to make them more suited for early recognition. More precisely, our framework is applicable to those methods that process segments of video sequentially [26,33,34]. We approach the early recognition problem from a novel viewpoint. To make the model recognize actions early, we make it biased toward early information of the video during the learning process.

When there is no earliness requirement, a learning algorithm may well take advantage of information from any part of the video, including the latter frames, while enough information may exist in the early parts of the video. However, in early recognition, the task is to classify the video when a small part of the video has been observed. Therefore, the algorithm must be able to take full advantage of the information residing in the early parts of the clip. To this end, we stop the learning process on a clip early, when the gained information from the clip has reached a sufficient amount, making the algorithm focused on the early parts. To quantify the amount of information learned from a frame, we make use of the loss value.

The rest of this article is organized as follows: In Sect. 2, we review related work, In Sect. 3, the proposed method is described, in Sect. 4, the experiments are presented, and Sect. 5 concludes the paper.

2 Related work

In this paper, we propose a framework applied to a kind of deep learning action recognition methods, called two-stream networks, making them more useful for the action prediction task. Hence, in Sect. 2.1 we review prediction methods and survey two-stream methods in Sect. 2.2.

2.1 Action prediction

The work by Ryoo [25] is one of the first attempts at early action recognition. They introduce two techniques called integral bag-of-words (IBOW) and dynamic bag-of-words (DBOW). They define per-class and per-progress-level action models and find the optimal pair of action and progress level pair using maximum likelihood. Integral bag-of-words works by computing histograms for various progress levels. DBOW computes the alignment between the query video and models, posing a temporal constraint on the alignment. After that, a dynamic programming approach is used to solve the alignment problem efficiently. In their work, each class and progress level model is obtained by averaging over the videos' histograms. When the number of videos is small or outliers are present, this method will be prone to errors. To better cope with these problems, [1] uses video feature vectors as the bases of a sparse coding representation. This representation is then used in place of the averaging strategy.

Some researchers follow a global-local paradigm, which means that they combine both global and local models for the early recognition task. The global component models a partial video considering all of its segments, while the local component models individual segments. In [12], the actual model is a linear combination of global and local models. Each model is a joint feature map, i.e. a function ψ of both the predictor x and the target variable y where $\psi(x, y)$ shows how likely the observation x is from class y. This work was further developed into a technique called Max-Margin Action Prediction Machine (MMAPM) [11] by adding composite kernels. Lai et al. [16] argue that as more parts of the video have been observed, the importance of segments (weights of local components) can vary, and they formulate the problem accordingly. They use a metric learning technique, which makes their method extensible to new action classes.

Wang et al. [32] take advantage of "mid-level" features for action prediction. First, they extract low-level features from the video and then cluster them into mid-level features called action units.

Some methods have been proposed based on deep learning. DeepSCN [13] generates a representation of videos

considering a few constraints: representations of partial videos are close to the complete video, representations within the same class are similar, and features are robust to noise. Layers of these representations are built on top of each other, leading to a deep network that is trained layer-wise. The authors improve their work in [14] (AAPNET), in which Generative Adversarial Networks (GANs) are used to classify videos. While in DeepSCN, the representations are learned independently of class labels, AAPNet learns the representation in the classifier's training, which improves the discrimination power. Another work based on GANs is AP-GAN [3]. It takes advantage of GANs for the prediction task. The system comprises two components: the action prediction module, where the GAN is used to predict (i.e., generate) future skeleton poses from the previous ones, and the action recognition module which classifies the sequence of poses.

Reference [20] uses Long Short-Term Memory (LSTM) networks for action prediction. They incorporate two constraints into their training process: that the correct class score is non-decreasing and that the margin between the score of the correct class and other classes is non-decreasing, too. Weng et al. [36] propose to use an LSTM for action anticipation and use an "agent" to exclude some of the categories as more parts of the video are observed. This agent is trained using reinforcement learning. Furnari and Farinella [6] propose the rolling-unrolling LSTM architecture which uses two LSTM networks. The rolling network encodes observed snippets, and the unrolling network predicts future representations.

SSNet [19] is a convolutional neural network over the temporal axis. The network jointly predicts both the action class and the temporal extent of the action. The scale is used to select the layer with a "proper" perception field, and then, the class label predicted by this layer is considered the recognition result. Hu et al. Hu et al. [9] propose the idea of soft regression. As partially observed videos are ambiguous, a soft label is used in training to model this uncertainty. This label is defined as a coefficient multiplied by the one-hot representation of the full video's label. This coefficient is learned jointly with the action predictor. A deep representation is used at the frame level, and the frame representations are connected in a fashion similar to recurrent neural networks (RNNs).

2.2 Two-stream networks

Simonyan and Zisserman [26] propose two-stream convolutional networks that process spatial and temporal streams by two separate networks. Parts of the video are given to these networks, and at the end, the results are aggregated. The input to the spatial network is single frames, while the input to the temporal network is stacks of motion images. A motion image can be either an optical flow or a trajectory image.

Reference [33] introduces Temporal Segment Network (TSN). The input video is partitioned into equal-length segments, and from each segment, a snippet (a short sequence of frames) is sampled and fed to the network. They also propose a cross-modality training, making it possible to use a CNN trained on still images for the temporal stream. Reference [34] extends Ref. [33] in several ways. For example, it introduces new aggregation schemes. Furthermore, it uses the TSN to untrimmed videos, i.e. where the onset of action in the video is unknown.

Reference [35] introduces a type of two-stream network called two-stream SR-CNNs (Semantic Region CNN). They replace the last pooling layer with a layer called region-of-interest (ROI) pooling. This layer separates proposed bounding boxes for different channels. The channels are human, object, and the scene, where human and objects have corresponding bounding boxes, and the scene simply corresponds to the whole frame. Then, each channel is sent to a separate network consisting of fully connected layers.

Reference [21] introduces a two-stream architecture based on Faster R-CNN (Region CNN) [24]. R-CNN produces RoI (region of interest) proposals and detects objects based on them. The architecture proposed by [21] produces the RoI proposals based on both appearance and motion information. An ROI fusion layer aggregates the ROIs of the two streams.

3 Proposed method

In this section, we describe our proposed method named early stopped learning in detail. First, we explain why it might be a good idea to stop training on a video early. Then, we propose a mechanism to decide at which point to discontinue training. Finally, an overview of the learning algorithm will be presented. We assume that many actions can be recognized before full execution is observed, and consequently, there must be discriminative information in the beginning parts of the video. We hypothesize that models with smaller recognition latency can be learned by focusing more on the beginning parts of the video in the training procedure.

In this paper, we split the video into equal-length segments, and from each segment sample a single frame. Thus, in the rest of the article, we use the terms frame and segment interchangeably. The idea of early-stopped learning is to only learn from a limited number of segments from the beginning of the video. This, however, poses the question of how does cropping out part of the video aid learning? To answer this question, note that all parts of the video may include useful information for classification, and this information may be redundant. When there is no limitation as to which parts of the video to learn from, the learner may prefer features that are more informative than others but lie in the late parts of the clip and ignore some features residing in the early parts.

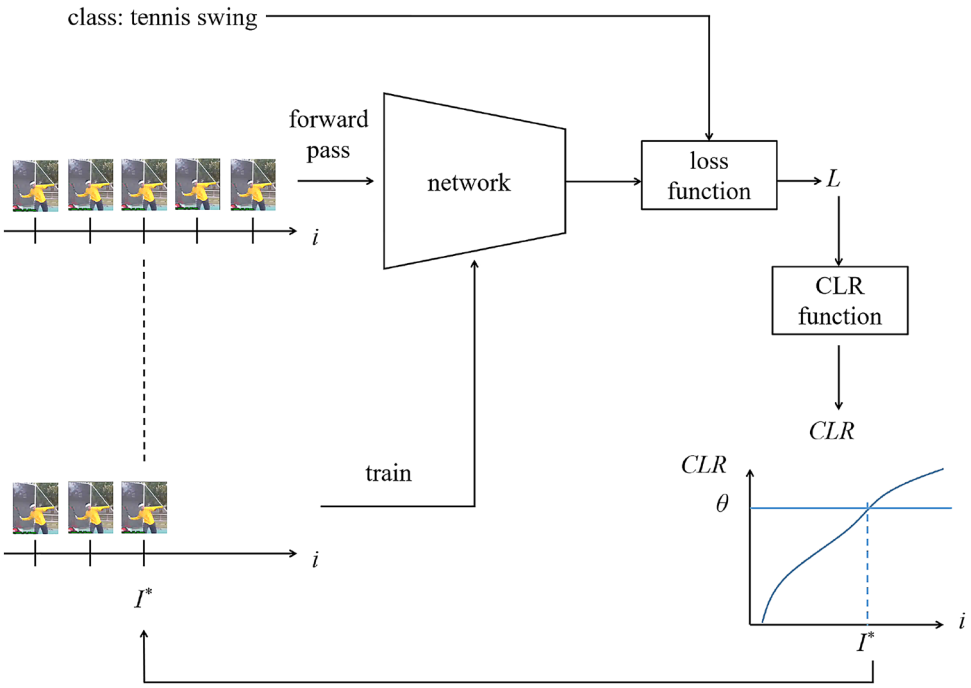

Fig. 1 Schema of the proposed method. A set of frames are passed through the network and their losses are computer. Then, based on the CLR value some of the frames are selected for training

This information may even be more helpful than other parts of the clip when there is no earliness requirement. However, when the learner is limited to the early parts, it is encouraged to take full advantage of the features residing in the early parts that might have gone unnoticed otherwise.

In video processing, there are several methods for applying deep networks. One of these methods is to use sequence models such as RNNs and LSTMs [9,20]. Another approach is to apply a convnet with 3D convolution operators [28]. Alternatively, the CNN architecture used in still image processing can be applied to video processing. This is done simply by applying the network to the individual frames or segments and then fusing the individual outputs (e.g. by averaging) [26,34].

In this paper, we deal with the last approach. In Temporal Segment Network (TSN) [34], a convnet is applied to a set of sampled frames, and the final result is computed by averaging the individual results (or using other types of fusion). In this way, the backpropagation update operation is effectively equal to back-propagate over the segments separately. However, for the early recognition task, our goal is to focus more on the beginning segments of the video. This is done by back-propagating over only the early portions of the video. In other words, we encourage our model to learn more from the beginning parts.

Neural networks use backpropagation to flow the information from training examples throughout the network and update its weights. The loss value is used to measure the discrepancy between the network's actual output and the correct output. By doing backpropagation on a training example, the loss value is expected to drop. Therefore, the loss value can be interpreted as "how much we can learn" by back-propagating on a particular example or "how far from perfect" we are at the current point. In other words, the loss can show how informative a training example (frame) can be.

The question is how many segments should be used for training. We define the cumulative loss ratio (CLR) at index i to be the sum of losses from the first segment to the i^{th} one, divided by the sum of losses of all segments, i.e.,

$$CLR_i = \sum_{j=1}^{i} L_j / \sum_{j=1}^{N} L_j \tag{1}$$

where N is the total number of segments.

This can be used to quantify how informative a partial video is. We can use this measure to decide when to stop learning on a video. In early-stopped learning, it is desired to only learn from a limited number of segments from the beginning of the video. We propose to stop learning when the CLR exceeds a threshold θ which is a hyper-parameter of the algorithm, i.e.

$$I^* = \arg\min_i \{1 \leq i \leq N \mid CLR_i \geq \theta\} \tag{2}$$

where I^* is the segment index up to which learning is performed. In summary, the algorithm for processing a video can be described as follows: we have a set of sampled frames for each video. The loss and CLR value are computed for each frame, and the value of I^* is computed based on (2). The frames 1 through I^* are used in training the network.

In practice, this operation is performed in mini-batches, and many frames can be processed in parallel. An overview of the procedure has been presented in Algorithm 1 and shown schematically in Fig. 1.

Input: $\boldsymbol{B}$: batch of videos; $\boldsymbol{W}$: model weights; η: learning rate; θ: threshold
Output: $\boldsymbol{W}$: updated model weights

$|\boldsymbol{B}|$ = number of videos in $\boldsymbol{B}$
for $k = 1$ **to** $|\boldsymbol{B}|$ **do**
  $\boldsymbol{V} = k^{th}$ video of $\boldsymbol{B}$
  N_k = number of segments of $\boldsymbol{V}$
  for $i = 1$ **to** N_k **do**
    L_{ki} = loss of i^{th} segment of $\boldsymbol{V}$
  end
  for $i = 1$ **to** N_k **do**
    $CLR_{ki} = \sum_{j=1}^{i} L_{kj} / \sum_{j=1}^{N_k} L_{kj}$
  end
  $I_k^* = \arg\min_i \{1 \leq i \leq N_k \mid CLR_{ki} \geq \theta\}$
end
$\boldsymbol{W} = \boldsymbol{W} - \eta \frac{1}{|\boldsymbol{B}|} \sum_{k=1}^{|\boldsymbol{B}|} \frac{1}{N_k} \sum_{j=1}^{I_k^*} \nabla_{\boldsymbol{W}} L_{kj}$

Algorithm 1: Processing of a mini-batch

4 Experiments

We use the Temporal Segment Network [34] at the core of our method. This architecture is of the two-stream network type [26], which means it consists of two convnets, one of which processes spatial and the other processes temporal stream. The input to the spatial network is one or more frames, and the input to the temporal network is a stack of frames, or optical flows, or trajectories. Various frames/segments are given separately to the network, and the outputs of the network for various frames/segments are combined. Finally, the results of the two networks are fused together.

TSN samples some snippets (a small set of frames) from the video as the input to the network. The network used for each stream (called base network) is a convnet which can be of any architecture such as the Resnet [8] or VGG [27] family. The authors propose to pre-train the network on ImageNet [4] data. The batch normalization used in the base network is frozen (i.e. no longer updated) after pre-training, except for the first layer. A dropout ratio of 0.8 and 0.7 is used for the spatial and temporal network, respectively. We used Resnet34 in our work, pre-trained on ImageNet. Another choice would be Resnet152, but we did not use it because of memory limitations. We only used the spatial component in our work even though adding the temporal part is trivial. The training process was run for 50 epochs, the learning rate was set to 0.001, and ten segments per video were used for training and 25 for testing.

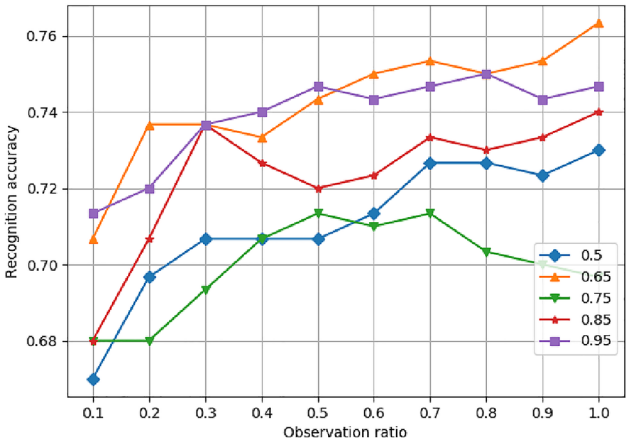

Fig. 2 Results of tuning the threshold parameter

The following datasets were used in the experiments: HMDB51 [15] which consists of 6849 videos divided into 51 classes, with each class consisting of at least 101 videos. The classes are of five general types: facial actions, facial actions with object manipulation, body movements, body movements with object interaction, and body movements for human interaction. The videos have been gathered from movies, YouTube clips, etc. The data have been partitioned into train and test sets in three different ways (called splits), available with the dataset. We ran experiments on each split and reported the average.

We use a small subset of HMDB51 for parameter tuning in this work, which we call SubHMDB. It consists of the ten following classes: brush hair, cartwheel, catch, chew, clap, climb, climb stairs, dive, draw sword, and dribble. Each class consists of 70 training and 30 test videos chosen randomly.

UCF11 [18] contains 1600 videos grouped into 11 categories, namely: basketball shooting, biking/cycling, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. The videos are challenging due to variations in camera motion, scale, etc. Each category is divided into 25 groups with more than four videos in each. Videos of each group share properties such as actor and background. We selected 70% of the data for training and the rest for testing with stratified sampling.

The algorithm has a hyper-parameter θ (2), the threshold for cumulative loss ratio which lies in the unit interval. We tested the algorithm on SubHMDB with values of θ selected from the set {0.3, 0.4, 0.5, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95}. The best result is achieved for $\theta = 0.65$. Figure 2 shows the results for some selected values of θ.

We compare our work with some of the recent methods proposed in the literature. These include GLTS [16], MTSSVM [12], SC [1], MSSC [1], and MMAPM [11]. As our method is an improvement upon TSN [34], we also com-

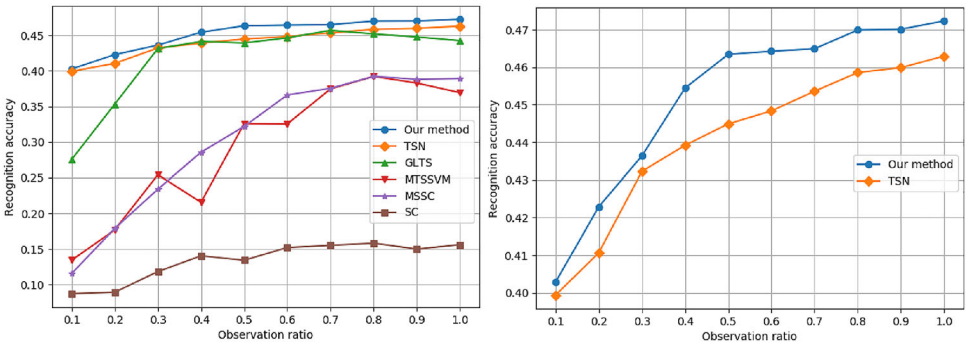

Fig. 3 Results on HMDB51. Left: compared with several baselines, including TSN, right: compared only with TSN

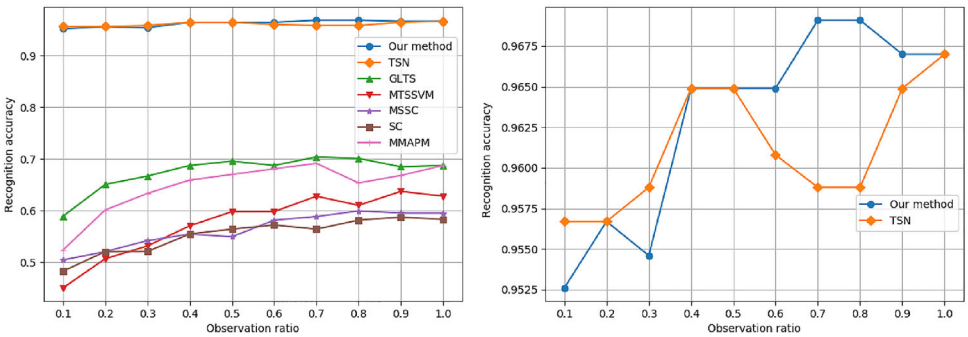

Fig. 4 Results on UCF11. Left: compared with several baselines, including TSN, right: compared only with TSN

pare our method with it. The difference between the two is that we use a subset of segments of TSN, selected from the beginning of the video as described in Sect. 3. To better highlight the difference with TSN, we include a separate graph for each dataset, including only TSN as the baseline.

Figure 3 shows the results on HMDB. Our method is of the highest accuracy, followed by TSN and then GLTS. The margin between our method and TSN is between 1 and 2% when the observation ratio is above 0.3. GLTS is the closest to our method among the baselines (except TSN). As can be seen, there is a big difference between our approach and GLTS when a small portion of the video has been observed (approximately 13% difference).

On UCF11, the two TSN-based methods (ours and basic TSN) outperform the other methods by a large margin of approximately 25%. These two methods have more than 95% accuracy and can be considered nearly "saturated." Our technique performs better than TSN, especially at greater observation ratios and performs slightly worse when the observation ratio is small. In all cases, the difference between them is around 1%. On average, our method is better. Like HMDB, the difference between our method and GLTS is more when the observation ratio is small. The results are illustrated in Fig. 4.

In summary, our method is a modification to the learning process of a deep learning-based action recognition method. The benefit is that it can be easily applied to similar action recognition methods to make them more suited to action prediction. The proposed method shows improvement compared to other methods. On the other hand, the disadvantage is that the enhancement of our technique compared to the baseline TSN is marginal.

5 Conclusion

In this paper, we presented a novel deep learning framework for early action recognition. It modifies how backpropagation is applied to individual frames in a deep learning algorithm, making the algorithm more suitable for early recognition. More specifically, it uses the loss value to measure how informative a frame is and stops learning at a frame when significant knowledge has been learned. We applied this framework to an action recognition model, called temporal segment network, on two well-known datasets UCF11 and HMDB51. Experimental results show that our method was able to boost the network's performance in the early recognition setup. We also compared our approach with some state-of-the-art methods, which showed that our method outperforms them.

Acknowledgements The authors would like to thank Dr. Mohsen Ramezani for reviewing the manuscript, and for his valuable comments.

References

1. Cao Y, Barrett D, Barbu A, Narayanaswamy S, Yu H, Michaux A, Lin Y, Dickinson S, Siskind JM, Wang S (2013) Recognize human activities from partially observed videos. In: Proceedings of the IEEE computer society conference on computer vision and pattern

recognition, pp 2658–2665. https://doi.org/10.1109/CVPR.2013.343
2. Chakraborty B, Holte MB, Moeslund TB, Gonzàlez J (2012) Selective spatio-temporal interest points. Comput Vis Image Underst 116(3):396–410. https://doi.org/10.1016/j.cviu.2011.09.010
3. Cui R, Hua G, Wu J (2020) AP-GAN: predicting skeletal activity to improve early activity recognition. J Vis Commun Image Represent 73:102923. https://doi.org/10.1016/j.jvcir.2020.102923
4. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 248–255
5. Dollár P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse spatio-temporal features. In: Proceedings - 2nd Joint IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance, VS-PETS, vol 2005, pp 65–72. https://doi.org/10.1109/VSPETS.2005.1570899
6. Furnari A, Farinella G (2020) Rolling-unrolling LSTMs for action anticipation from first-person video. IEEE Transactions on Pattern Analysis and Machine Intelligence, p 1. https://doi.org/10.1109/tpami.2020.2992889
7. Harris CG, Stephens (1988) A combined corner and edge detector. In: Alvey vision conference, vol 15, pp 189–192
8. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778. www.image-net.org
9. Hu JF, Zheng WS, Ma L, Wang G, Lai JH, Zhang J (2018) Early action prediction by soft regression. IEEE Trans Pattern Anal Mach Intell 41(11):2568–2583. https://doi.org/10.1109/TPAMI.2018.2863279
10. Kantorov V, Laptev I (2014) Efficient feature extraction, encoding, and classification for action recognition. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, pp 2593–2600. https://doi.org/10.1109/CVPR.2014.332
11. Kong Y, Fu Y (2016) Max-margin action prediction machine. IEEE Trans Pattern Anal Mach Intell 38(9):1844–1858. https://doi.org/10.1109/TPAMI.2015.2491928
12. Kong Y, Kit D, Fu Y (2014) A discriminative model with multiple temporal scales for action prediction. In: Fleet D et al (eds) ECCV 2014, Part V, LNCS 8693, Springer. pp. 596–611. https://doi.org/10.1007/978-3-319-10602-1_39
13. Kong Y, Tao Z, Fu Y (2017) Deep sequential context networks for action prediction. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 3662–3670. https://doi.org/10.1109/CVPR.2017.390. http://ieeexplore.ieee.org/document/8099873/
14. Kong Y, Tao Z, Fu Y (2018) Adversarial action prediction networks. IEEE Trans Pattern Anal Mach Intell 42(3):539–553
15. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T (2011) HMDB: a large video database for human motion recognition. In: Proceedings of the IEEE international conference on computer vision, pp 2556–2563. https://doi.org/10.1109/ICCV.2011.6126543
16. Lai S, Zheng WS, Hu JF, Zhang J (2017) Global-local temporal saliency action prediction. IEEE Trans Image Process 27(5):2272–2285. https://doi.org/10.1109/TIP.2017.2751145
17. Laptev Li (2003) Space–time interest points. In: Proceedings ninth IEEE international conference on computer vision, pp 432–439. https://doi.org/10.1109/ICCV.2003.1238378
18. Liu J, Luo J, Shah M (2009) Recognizing realistic actions from videos in the Wild. In: 2009 IEEE computer society conference on computer vision and pattern recognition workshops, CVPR workshops 2009, pp 1996–2003. https://doi.org/10.1109/CVPRW.2009.5206744
19. Liu J, Shahroudy A, Wang G, Duan LY, Kot AC (2018) Ssnet: scale selection network for online 3d action prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8349–8358
20. Ma S, Sigal L, Sclaroff S (2016) Learning activity progression in LSTMs for activity detection and early detection. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 1942–1950. https://doi.org/10.1109/CVPR.2016.214. http://ieeexplore.ieee.org/document/7780583/
21. Peng X, Schmid C (2016) Multi-region two-stream R-CNN for action detection. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), pp 744–759. https://doi.org/10.1007/978-3-319-46493-0_45
22. Qiao R, Liu L, Shen C, van den Hengel A (2017) Learning discriminative trajectorylet detector sets for accurate skeleton-based action recognition. Pattern Recogn 66:202–212. https://doi.org/10.1016/j.patcog.2017.01.015
23. Ramezani M, Yaghmaee F (2016) A review on human action analysis in videos for retrieval applications. Artif Intell Rev 46(4):485–514. https://doi.org/10.1007/s10462-016-9473-y
24. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 2164–2173
25. Ryoo MS (2011) Human activity prediction: early recognition of ongoing activities from streaming videos. In: Proceedings of the IEEE international conference on computer vision, pp 1036–1043. https://doi.org/10.1109/ICCV.2011.6126349
26. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, vol 1. Neural information processing systems foundation, pp 568–576
27. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: 3rd international conference on learning representations, ICLR 2015 - Conference Track Proceedings
28. Tran D, Wang H, Torresani L, Ray J, Lecun Y, Paluri M (2018) A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, pp 6450–6459. https://doi.org/10.1109/CVPR.2018.00675. http://openaccess.thecvf.com/content_cvpr_2018/html/Tran_A_Closer_Look_CVPR_2018_paper.html
29. Wang H, Kläser A, Schmid C, Liu CL (2011) Action recognition by dense trajectories. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, pp 3169–3176. https://doi.org/10.1109/CVPR.2011.5995407
30. Wang H, Kläser A, Schmid C, Liu CL (2013) Dense trajectories and motion boundary descriptors for action recognition. Int J Comput Vis 103(1):60–79. https://doi.org/10.1007/s11263-012-0594-8
31. Wang H, Schmid C (2013) Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision, pp 3551–3558. https://doi.org/10.1109/ICCV.2013.441
32. Wang H, Yuan C, Shen J, Yang W, Ling H (2018) Action unit detection and key frame selection for human activity prediction. Neurocomputing 318:109–119. https://doi.org/10.1016/j.neucom.2018.08.037
33. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, van Gool L (2016) Temporal segment networks: towards good practices for deep action recognition. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol 9912 LNCS, pp 20–36. https://doi.org/10.1007/978-3-319-46484-8_2

34. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2018) Temporal segment networks for action recognition in videos. IEEE Trans Pattern Anal Mach Intell 41(11):2740–2755
35. Wang Y, Song J, Wang L, Gool L, Hilliges O (2016) Two-stream SR-CNNs for action recognition in videos. In: Proceedings of the British machine vision conference (BMVC), pp 108.1–108.12. https://doi.org/10.5244/c.30.108
36. Weng J, Jiang X, Zheng WL, Yuan J (2020) Early action recognition with category exclusion using policy-based reinforcement learning. IEEE Trans Circuits Syst Video Technol, p 1. https://doi.org/10.1109/tcsvt.2020.2976789
37. Zanfir M, Leordeanu M, Sminchisescu C (2013) The moving pose: an efficient 3D kinematics descriptor for low-latency action recognition and detection. In: Proceedings of the IEEE international conference on computer vision, pp 2752–2759. https://doi.org/10.1109/ICCV.2013.342
38. Zhang HB, Zhang YX, Zhong B, Lei Q, Yang L, Du JX, Chen DS (2019) A comprehensive survey of vision-based human action recognition methods. Sensors 19(5):1005. https://doi.org/10.3390/s19051005