Towards Streaming Egocentric Action Anticipation

Antonino Furnari
Department of Mathematics and Computer Science
University of Catania, Catania, Italy
Next Vision s.r.l., Catania, Italy
Email: antonino.furnari@unict.it

Giovanni Maria Farinella
Department of Mathematics and Computer Science
University of Catania, Catania, Italy
Next Vision s.r.l., Catania, Italy
Email: giovanni.farinella@unict.it

Abstract—Egocentric action anticipation consists in predicting the future actions a camera wearer will likely perform based on past video observations. While in a real-world system it is fundamental to produce such predictions before the action begins, past works have not generally paid attention to model runtime during evaluation. Indeed, current evaluation schemes assume that predictions can be made offline, and hence that computational resources are not limited. In this paper, we propose a “streaming” evaluation protocol which explicitly considers model runtime for performance assessment, assuming that predictions will be available only after the current video segment is processed, which depends on the processing time of a method. Following the proposed evaluation scheme, we benchmark different state-of-the-art approaches for egocentric action anticipation on two popular datasets. Our analysis shows that models with a smaller runtime tend to outperform heavier models in the considered streaming scenario, thus changing the rankings observed in standard offline evaluations. Based on this observation, we propose a lightweight action anticipation model consisting in a simple feed-forward 3D CNN, which we propose to optimize using knowledge distillation techniques and a custom loss. The results show that the proposed approach outperforms prior art in the streaming scenario, also in combination with other lightweight models.

1. INTRODUCTION

Egocentric action anticipation, which consists in predicting a plausible future action before it is performed from egocentric video, has recently attracted a lot of attention [1], [2], [3], [4], [5], [6], [7], [8], [9]. To make sure that action anticipation is practically useful, the predictions about the future should be available before the action is initiated by the user. Nevertheless, previous works have not considered the computational time required to process the input video, assuming a negligible runtime [2], [4], [7], [5], [8], [9], [6]. We argue that this assumption can lead to unfair and overly optimistic evaluations and propose that egocentric action anticipation should be evaluated considering a “streaming” scenario. In these settings, the anticipation algorithm processes video segments as they become available and outputs predictions after the computational time required by the anticipation model is elapsed. Figure 1 reports different schemes to evaluate egocentric action anticipation. In the scheme, an action anticipation model is used to predict the label of a future action (in green). The model has been designed to anticipate actions beginning in \( \tau_a \) seconds (target anticipation time) observing a video segment of \( \tau_r \) seconds (observation time). Current approaches implicitly assume that the model runtime \( \tau_r \) is negligible \( (\tau_r = 0) \) and so the prediction for a given video segment will be available right after it is passed to the model (Figure 1(a)). However, in a realistic case in which the model runtime is larger than zero \( (\tau_r > 0) \), predictions about future actions will be available at a later timestamp, which makes the effective anticipation time \( (\hat{\tau}_a) \) smaller than the target one \( (\tau_a) \) (Figure 1(b)). Since different methods are likely characterized by different runtimes, they will also have different effective anticipation times \( \hat{\tau}_a \), which can make comparisons under the classic scheme limited. To evaluate methods more uniformly, the temporal bounds of the observed video can be adjusted at test time to make the effective anticipation time equal to or larger than the target one: \( \hat{\tau}_a \geq \tau_a \). As illustrated in Figure 1(c), this can be obtained by shifting the observed video backwards by \( \tau_r \) seconds, hence sampling input observed videos in advance.

In this paper, we propose a new evaluation scheme which allows to assess the performance of existing egocentric action anticipation approaches in a streaming scenario by recomputing the timestamps at which input video segments are sampled at test time depending on the target anticipation time, the method’s estimated runtime, and the length of the video observation. Since in this scenario smaller runtimes are more effective, we propose a lightweight egocentric action anticipation model based on simple feed-forward 3D CNNs. While feed-forward 3D CNNs are fast, we note that their performance tends to be limited as compared to full-fledged models involving different components and multi-modal observations. We hence propose to optimize the performance of such models by using a future-to-past knowledge distillation approach which allows to transfer the knowledge from an action recognition model to the target anticipation network.

Fig. 1. Different schemes to evaluate egocentric action anticipation methods. See text for discussion.
Experiments on two popular datasets, EPIC-KITCHENS-55 and EGTEA Gaze+, show that 1) the proposed evaluation scheme induces a different ranking over state-of-the-art methods with respect to non-streaming evaluation schemes, which suggests that current evaluation protocols could be biased and incomplete, 2) the proposed lightweight method based on 3D CNNs achieves state-of-the-art results in the streaming scenario, which advocates for the viability of knowledge distillation to make learning more data-effective for streaming action anticipation, 3) lightweight approaches tend to outperform more sophisticated methods requiring different modalities and complex feature extractors in the streaming scenario, which suggests that more attention should be paid to runtime optimization when tackling anticipation tasks.

II. RELATED WORK

a) Efficient, Online and Streaming Vision: Works in image and video understanding generally assumed that computation can be performed offline [10], [11], [12], [13], [14], [15], [16], [17], [18], [14], [4], [7], [5]. Hence, little attention has been paid by these methods to model runtime. Another line of research has devoted its attention to the development of computationally efficient models for image recognition [19], [20], [21], object detection [22], [23], [24] and action recognition [25]. These works have usually been evaluated considering standard evaluation schemes, as well as reporting model runtime or number of FLoating Operations Per Second (FLOPS). Previous works have also tackled the problem of online video processing, studying online action detection [26], early action recognition [27], [28], action anticipation [2], [29], [4] and next active object prediction [30]. These tasks require the developed algorithm to make predictions before a video event is completely or even partially observed. However, they generally do not evaluate models according to runtime, hence assuming computational resources not to be finite. Some works have recently considered a “streaming” scenario in which algorithms should be evaluated considering the time in which the predictions are available depending on model runtime [31], [32].

We build on works on efficient computer vision [25], [20], online video processing [26] and streaming perception [32]. However, differently from these works, we study the streaming scenario within the task of egocentric action anticipation, in which the timeliness of predictions is fundamental to ensure their practical utility. To this aim, we propose an evaluation scheme which explicitly takes into account model runtime and the streaming nature of video, allowing to compare algorithms in a more homogeneous and practical way.

b) Egocentric Action Anticipation: Different works have tackled this task [4], [7], [3], [5]. Previous approaches have considered baselines designed for action recognition [2], defined custom losses [33], modeled the evolution of scene attributes and action over time [34], disentangled the tasks of encoding and anticipation [4], aggregated features over time [7], predicted motor attention [5], leveraged contact representations [3], mimicked intuitive and analytical thinking [9], and predicted future representations [8]. While these approaches have been designed to maximize performance when predicting the future, they have never been evaluated in a streaming scenario. In fact model runtime has not been reported or even mentioned in past works.

We benchmark some representatives of these methods and show experimentally that their performance is limited in the streaming scenario. We contribute a novel streaming evaluation scheme which can be used to assess performance considering a more realistic and practical scenario and propose a lightweight model for egocentric action anticipation which can be used in resource-constrained settings.

c) Knowledge Distillation: Knowledge distillation techniques aim to transfer the knowledge from a large and expensive neural network (the teacher) to a small and lightweight network (the student). Initial approaches used the logits of the teacher model as a supervisory signal for the student [35]. These methods aimed to minimize the divergence between the probability distributions predicted by the student and the teacher for a given input example, with the goal of providing a richer learning objective as compared to deterministic ground truth labels. Other approaches used the activations of the intermediate layers of the teacher to guide the optimization of the student [36], [37], whereas some methods modeled the relationships between the outputs of different layers of the network to provide guidance to the student on how to process the input example [38], [39].

We use knowledge distillation to optimize the performance of a lightweight action anticipation model and achieve competitive results in the streaming scenario. Differently from the aforementioned works, we do not transfer knowledge from a complex model to a simple one. Instead, we use an action recognition model looking at a future action as the teacher, and a lightweight model looking at a past video snippet as the student. This distillation approach is referred to in this paper as “future-to-past”. We propose a loss which facilitates knowledge transfer using the teacher to instruct the student on which spatiotemporal features are more discriminative for action anticipation.

d) Knowledge Distillation for Future Prediction: Few previous works used knowledge distillation to improve action anticipation. Some works used knowledge distillation implicitly by encouraging the model to predict representations of future frames, which were later used to make the predictions [40], [29], [8]. Other works have explicitly considered distillation approaches to transfer knowledge from a fixed teacher with label smoothing [1] or from an action recognition model [41], [42]. Other works applied knowledge techniques to early action prediction [43].

Similarly to these works, we study methods to transfer knowledge from an action recognition model to the target anticipation network. However, differently from previous works, our main objective is to assist the optimization of a lightweight and computationally efficient model.
III. STREAMING EVALUATION SCHEME

Let $V$ be the input video, and let $V_{t_s:t_e}$ denote a video clip starting at timestamp $t_s$ and ending at timestamp $t_e$. Let $\phi$ be the model designed to process videos of length $\tau_o$ (observation time) and anticipate actions happening after $\tau_o$ seconds (anticipation time). At a given timestamp $t$, the algorithm processes the most recent video segment $V_{t-\tau_o:t}$. Let $\tau_r$ be the time required by the model to process a video segment and output a prediction (model runtime). The prediction $\phi(V_{t-\tau_o:t})$ will be available at timestamp $t + \tau_r$. We hence denote it as $\hat{y}_{t+\tau_r} = \phi(V_{t-\tau_o:t})$. Similarly, we denote a prediction available at time $t$ as $\hat{y}_t = \phi(V_{t-\tau_o:t-\tau_r:t-\tau_r})$. This notation makes it explicit that, in the presence of a large runtime $\tau_r$, the input video segment should be sampled ahead of time to preserve a correct anticipation time. We assume that resources are limited and only a single GPU process is allowed at a time, which is a realistic scenario when algorithms are deployed to a wearable device. Note that, in this case, the runtime will not be negligible ($\tau_r > 0$), and hence the model will not be able to make predictions at every single frame of the video. Specifically, predictions will be available only at selected timestamps $t_o + k \cdot \tau_r, k \in \mathbb{N}^+$. This happens because the model first needs to wait for $\tau_o$ seconds to fill the video buffer, then it has to wait for the runtime $\tau_r$ to make the next prediction. As a consequence, there will be a difference between the ideal video segment in offline settings and the one actually sampled when processing the video in streaming mode. Figure 2(a) illustrates an example of the video sampling introduced by the proposed streaming action anticipation scenario and the related difference between ideal and real observed video segments.

For consistency with past benchmarks and to use the labels available in current action anticipation datasets, we evaluate anticipation methods only at specific timestamps sampled $\tau_a$ seconds before the beginning of each action. In particular, let $A_i = (s_i, c_i, y_i)$ be the $i$th labeled action of a test video $V$, where $s_i$ denotes the action start timestamp, $c_i$ denotes the action end timestamp, and $y_i$ denotes the action label. The action $A_i$ will be associated to the most recent prediction made $\tau_a$ seconds before the beginning of the action, at timestamp $s_i - \tau_a$. It is worth noting that a prediction might not be available exactly at the required timestamp (see Figure 2(a)), so we will consider the most recent timestamp at which a prediction is available, which is given by the following formula:

$$t_i^* = \left\lfloor \frac{s_i - \tau_a - \tau_o}{\tau_r} \right\rfloor \cdot \tau_r + \tau_o - \tau_r. \tag{1}$$

Figure 2(b) reports a scheme of the formula. The term $\left\lfloor \frac{s_i - \tau_a - \tau_o}{\tau_r} \right\rfloor \cdot \tau_r$ computes the number of time slots of length $\tau_r$ completely included in the segment of length $s_i - \tau_a - \tau_o$ ((i) in Figure 2(b)). Subtracting $\tau_o$ is necessary because the first video segment will be processed only after $\tau_o$ seconds. The product $\left\lfloor \frac{s_i - \tau_a - \tau_o}{\tau_r} \right\rfloor \cdot \tau_r$ ((ii) in Figure 2(b)) quantizes the timestamp. The term $\tau_o$ adds the observation time which had been previously subtracted ((iii) in Figure 2(b)) and the term $-\tau_r$ subtracts the runtime needed to obtain the prediction ((iv) in Figure 2(b)). We will hence associate the following prediction to action $A_i$: $\hat{y}_{t_i^*} = \phi(V_{t_i^*-\tau_o:t_i^*})$.

IV. METHOD

Based on the observation that model runtime influences performance in the considered streaming evaluation scheme, we propose a lightweight egocentric action anticipation approach based on simple feed-forward 3D CNNS. Differently from full-fledged anticipation models involving multi-modal feature extraction and sequence processing [4], [7], [29], feed-forward 3D CNNS are simple and fast, but also harder to optimize for action anticipation. To tackle this issue, we propose a training scheme based on knowledge distillation which we show to greatly improve performance of feed-forward 3D CNNS.

a) Proposed Training Scheme: Figure 3 illustrates the proposed training procedure, which operates over both labeled and unlabeled examples. Given a video $V$ and a timestamp $t$, we form a training example considering a pair of videos including a past observation $V_{t-\tau_o:t-\tau_r:t-\tau_r}$, a future observation $V_{t:t+\tau_r}$, and a label $y_t$ denoting the class of the action included in the future observation $V_{t:t+\tau_r}$. Depending on the sampled timestamp $t$, the future video segment might not be associated to any action, in which case we will say that the example is unlabeled and denote its label with $y_t = \mathbb{O}$. As a teacher, we use a 3D CNN $\phi^T$ pre-trained to perform action recognition from input videos of resolution $C \times F \times H \times W$, where $C$ is the number of channels ($C = 3$ for RGB videos), $F$ is the number of frames, $H$ and $W$ are the video frame height and width respectively. The teacher $\phi^T = \beta^T \circ \gamma^T$ is composed of a backbone $\beta^T$ which extracts spatio-temporal representations of resolution $C_r \times F_r \times H_r \times W_r$, as well as a classifier $\gamma^T$ which predicts a probability distribution over classes. The student model $\phi^S = \beta^S \circ \gamma^S$ has the same structure as the teacher $\phi^T$ and it is initialized with the same
weights. We train the model by feeding past observations to the student and future observations to the teacher. At each training iteration, we extract the internal representation of the past segment \(r_t^p\) = \(\beta^S(V_{t-\tau a, t-\tau o})\), the representation of the paired future segment \(r_t^f\) = \(\beta^T(V_{t+\tau o, t+\tau a})\), and the predicted future action label \(\hat{y}_t = \gamma^T(r_t^f)\).

We hence train the student to both classify the past observation correctly and extract representations coherent with the ones of the future segment using the following loss:

\[
\mathcal{L} = \lambda_d \mathcal{L}_d(r_t^p, r_t^f) + \|y_t \neq \emptyset\| \lambda_c \mathcal{L}_c(y_t, \hat{y}_t),
\]

(2)

where \(\mathcal{L}_d\) is a distillation loss used to encourage the representations of the past and future segments to be coherent, \(\mathcal{L}_c\) is a classification loss which aims to reduce the classification error of the student (e.g., cross entropy loss), \(\lambda_d\) and \(\lambda_c\) are hyper-parameters used to regulate the contributions of the two losses, and the Iverson bracket term \([y_t \neq \emptyset]\) is used to avoid computing the classification loss when the example is unlabeled.

b) Proposed Knowledge Distillation Loss: Unlike classic knowledge distillation, in our training protocol, the teacher and student models process different but related inputs (i.e., the past and future video segments). Let \(r_t^p(i)\) and \(r_t^f(i)\) be the \(C_i\)-dimensional temporal locations of the \(r_t^p\) and \(r_t^f\) tensors. One way to enforce knowledge distillation would be to maximize the similarity between corresponding representations \(r_t^p(i)\) and \(r_t^f(i)\) directly, e.g., using the Mean Squared Error (MSE) loss. However, since the inputs of the two networks are different, we expect their representations to be spatio-temporally misaligned. To mitigate this misalignment, we propose to maximize similarity between all \((r_t^p(i), r_t^f(j))\) pairs \(\forall i, j \in 1, \ldots, F_r \cdot H_r \cdot W_r\). Practically, we define a future-to-past similarity matrix \(M\) as follows:

\[
M_{ij} = \frac{r_t^p(i) \cdot r_t^f(j)}{|r_t^p(i)||r_t^f(j)|}. \tag{3}
\]

Note that the general term of the matrix \(M_{ij}\) is the cosine similarity between the two representations \(r_t^p(i)\) and \(r_t^f(j)\). We hence maximize the values of \(M\) by minimizing the following loss:

\[
\mathcal{L}_d(r_t^p, r_t^f) = \min \left\{ \frac{1}{(F_r H_r W_r)^2} \sum_{j=1}^{F_r H_r W_r} M_{ij} \right\}^{-1}. \tag{4}
\]

The main rationale behind this training objective is to encourage the student network to pay attention to spatio-temporal locations in the past observation which contain semantic content important for action recognition in the future, even if these are not spatio-temporally aligned. For example, if the future action is "take plate", the student network should learn to extract "plate" features from the past segment even if the features occur at different spatio-temporal regions. Figure 4 illustrates the proposed loss.

V. EXPERIMENTAL SETTINGS

We perform experiments on two datasets: EPIC-KITCHENS-55 [2] and EGTEA Gaze+ [44]. EPIC-KITCHENS-55 [2] is composed of 432 videos acquired by 32 subjects and labeled with 39,595 action segments with a taxonomy of 125 verbs and 352 nouns. We split the public dataset following [4] to obtain training set of 232 videos and 23,493 action segments, and a validation set of 40 videos and 4,979 segments. We report results on the validation set. We consider all unique \((verb, noun)\) pairs appearing in the public set to obtain 2,513 distinct action classes. EGTEA Gaze+ [44] includes 86 videos acquired by 28 different subjects, labeled with 10,325 action segments including 19 verbs, 51 nouns and 106 action classes. We randomly split...
the dataset into a training set containing 65 videos and a test set containing 21 videos. We report results on the test set.  

We benchmark different egocentric action anticipation approaches using both a classic offline evaluation scheme and the proposed streaming protocol described in Section III. We choose different approaches spanning from full-fledged but computationally expensive methods to the lightweight baselines:

- **Full-fledged methods**: We consider the Rolling-Unrolling LSTMs (RULSTM) proposed in [4], and an Encoder-Decoder (ED) architecture based on [29]. These approaches make use of LSTMs and consider different input modalities including representations extracted from RGB frames, optical flow, and object-based features. Due to the multi-modal data extraction and processing, these approaches are accurate but computationally expensive.

- **Action recognition baselines**: We consider methods achieving state-of-the-art results in action recognition. Specifically, we include different 3D feed-forward CNNs of varying computational complexity: a computationally expensive I3D [17] network processing clips of resolution $3 \times 64 \times 224 \times 224$, the more efficient SlowFast [18] network based of ResNet50 with inputs of resolution $3 \times 32 \times 224 \times 224$, the computationally optimized X3D-XS architecture [25] processing inputs of size $3 \times 4 \times 160 \times 160$, and the lightweight R(2+1)D [45] CNN based on ResNet18 and processing videos of resolution $3 \times 16 \times 64 \times 64$.

- **Lightweight baselines**: We also include two baselines designed to be very lightweight, but likely to be less accurate. In particular, we consider Temporal Segment Networks (TSN) [46] and a simple LSTM [47] which takes as input representations of the RGB frames obtained using a BNInception 2D CNN pre-trained for action recognition.

See the supplementary material for implementation details.

VI. RESULTS

A. Proposed Streaming Benchmark

Table I and Table II report the performances of the proposed methods on the considered datasets using Mean Top-5 Recall [33]. Results are reported using both a classic offline evaluation protocol (columns 4–6) and the proposed streaming evaluation scheme (columns 7–9). Column 2 of Table I also reports the runtime in milliseconds (R.TIME) whereas column 3 reports the framerate if frames per second (FPS). Verb, noun and action (ACT.) performance is reported using Mean Top-5 Recall%. Best results per column are reported in bold, whereas second-best results are underlined.

a) **EPIC-KITCHENS-55**: The results reported in Table I highlight how methods with large runtimes are more optimized for performance and tend to outperform competitors in the offline to streaming evaluation scenario. For example, ED is surpassed by RULSTM, which is also based on LSTMs but makes use of object-based features, which have a significant impact on runtime (ED has a runtime of 100.56ms, whereas RULSTM has a runtime of 724.98ms). Likewise, I3D outperforms SlowFast and X3D-XS probably due to the larger input size, at the cost of a larger runtime (275.26ms of I3D vs 173.73ms of SlowFast and 142.5ms of X3D X5). R(2+1)D is the lightest method among the considered 3D CNNs, with a framerate of 24.15fps, but achieves limited performance (8.10 offline action performance). In general, methods based on simple 3D CNNs tend to perform worse than highly optimized methods such as RULSTM. Notably, the proposed approach based on knowledge distillation improves the offline action performance of the R(2+1)D baseline by +4.73 (12.83 of DIST-R(2+1)D vs 8.10 of R(2+1)D-M).

It can be noted that runtime significantly affects performance when the streaming evaluation scenario is considered. Indeed, by comparing columns 6 and 9, it is clear that, while the performances of all methods decrease when passing from offline to streaming evaluation (compare columns 6 and 9), methods characterized by smaller runtimes are characterized by a smaller performance gap. For example, RULSTM has an offline action performance of 14.81, which is reduced to 11.40 in the streaming scenario (~3.41), whereas TSN, which is the fastest approach, has an offline action performance of 10.28 which is very comparable to the performance of 10.25 achieved in the streaming scenario (only ~0.03). Interestingly, the performance of the lightweight LSTM approach (runtime of 25.96ms) is barely affected when passing from an offline to an online evaluation scheme (12.35 offline action performance vs 12.18 streaming action performance).

The last row of Table I reports the performance of a method obtained by combining the LSTM with the proposed model based on knowledge distillation. Results point out that the obtained LSTM+DIST-R(2+1)D achieves good results at a small computational cost. Indeed, LSTM+DIST+R(2+1)D achieves a similar streaming action performance of 13.55, which significantly outperform other approaches such as RULSTM in the streaming scenario while running much faster with a runtime of 67.37ms and a framerate of 14.84fps.

It is worth noting that the offline and streaming evaluation standard offline evaluation scenario.
of the observed and future segments are the same, the example is discarded. 3) All - we consider all possible video clips, both labeled and unlabeled. These settings can be considered only when knowledge distillation techniques are employed.

Table III reports the results. It can be noted that just adding knowledge distillation techniques while keeping the amount of training data fixed always allows to improve performance. This suggests that knowledge distillation has a guiding effect which improves generalization. For instance, training the model with knowledge distillation relying only on supervised data allows to obtain a +2.45 in action performance when compared to training without knowledge distillation (10.55 vs 8.10). Simply augmenting the number of training examples has a similar regularizing effect. Indeed, training the model with the augmented data allows to obtain an action score of 10.66, which is larger by +2.46 with respect to the 8.10 obtained considering only supervised data. Combining both approaches allows to boost performance by +4.73 with respect to the base model (12.83 using all data and knowledge distillation vs 8.10 using only supervised data and no knowledge distillation). These results highlight how knowledge distillation has the effect of making training more data-efficient, allowing to increase the number of training examples and better exploit the available data.

VII. CONCLUSION

We presented a novel evaluation protocol for egocentric action anticipation which explicitly considers a streaming scenario in which the timeliness of predictions is crucial. We benchmarked different anticipation approach and shown that classic offline evaluations are limited when algorithms need to be deployed on real hardware. Noting that a small runtime is crucial for streaming anticipation, we proposed a method based on a lightweight 3D CNN and optimized with knowledge distillation techniques which achieved state-of-the-art performance on two datasets. We believe that the proposed investigation can be useful in practical scenarios in which the evaluated models need to be deployed on real hardware.

ACKNOWLEDGMENT

This research has been supported by MIUR AIM - Attrazione e Mobilitá Internazionale Linea 1 - AIM1893589 - CUP: E64118002540007, by MISE - PON I&C 2014-2020 - Progetto ENIGMA - Prog n. F/190050/02/X44 – CUP: B61B19000520008, and by the project MEGABIT - PIAno di inCenitivi per la Ricerca di Ateneo 2020/2022 (PIACERI) – linea di intervento 2, DMI - University of Catania.
REFERENCES

[1] Guglielmo Camporese, Pasquale Coscia, Antonino Furnari, Giovanni Maria Farinella, and Lamberto Ballan. Knowledge distillation for action recognition via label smoothing. In International Conference on Pattern Recognition (ICPR), 2020.

[2] Dima Daukantas, Hazel Doughty, Giovanni Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1–2, 2020.

[3] Eashom Dessalene, Chimmya Devaraj, Michael Mayord, Cornelia Ferruller, and Yiannis Alonimou. Forecasting action through contact representations from first-person video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

[4] Antonino Furnari and Giovanni Farinella. Rolling-unrolling lstms for action anticipation from first-person video. IEEE transactions on pattern analysis and machine intelligence, 2020.

[5] Miao Liu, Suyi Tang, Yi Li, and James M Rehg. Forecasting human-object interaction: joint prediction of motor attention and actions in first-person video. In European Conference on Computer Vision, pages 704–721. Springer, 2020.

[6] Zhaobo Qi, Shuhuai Wang, Chi Su, Li Su, Qingming Huang, and Qi Tian. Self-regulated learning for egocentric video activity anticipation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[7] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In European Conference on Computer Vision, pages 154–171. Springer, 2020.

[8] Yu Wu, Linchao Zhu, Xiaohan Wang, Yi Yang, and Fei Wu. Learning to anticipate egocentric actions by imagination. IEEE Transactions on Image Processing, 30:1143–1152, 2020.

[9] Tianyu Zhang, Weiqing Min, Ying Zhu, Yong Rui, and Shuangqiang Jiang. An egocentric action anticipation framework via fusing intuition and analysis. In Proceedings of the 28th ACM International Conference on Multimedia, pages 402–410, 2020.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations, 2014.

[12] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. GoogLeNet: Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015.

[13] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In European conference on computer vision, pages 525–542. Springer, 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9):1904–1916, 2015.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017.

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6):1137–1149, 2017.

[17] João 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, 2017.

[18] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6202–6211, 2019.

[19] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning, pages 6105–6114. PMLR, 2019.

[20] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[21] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size. arXiv preprint arXiv:1602.07360, 2016.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016.

[23] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7263–7271, 2017.

[24] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 203–213, 2020.

[25] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In European Conference on Computer Vision, pages 269–284. Springer, 2016.

[26] Minh Hoai and Fernando De la Torre. Max-margin early event detectors. In International Journal of Computer Vision, 107(2):191–202, 2014.

[27] Mohammad Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Basura Fernando, Lars Petersson, and Lars Andersson. Encouraging lstms to anticipate actions very early. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 280–289, 2017.

[28] Jiuyang Gao, Zhenheng Yang, and Ram Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. In British Machine Vision Conference, 2017.

[29] Antonino Furnari, Sebastiano Battiato, Kristen Grauman, and Giovanni Maria Farinella. Next-active-object prediction from egocentric videos. Journal of Visual Communication and Image Representation, 49:401–411, 2017.

[30] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka ˇCehovin Zajc, Tomas Vojir, Gustav Hager, Alan Lukezic, Abdelrahman Eldesokey, et al. The visual object tracking vot2017 challenge results. In Proceedings of the IEEE international conference on computer vision workshops, pages 1949–1972, 2017.

[31] Mengtian Li, Yu-Xiong Wang, and Deva Ramanan. Towards streaming perception. In European Conference on Computer Vision, pages 473–488. Springer, 2020.

[32] Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, pages 0–4, 2018.

[33] Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 0–0, 2019.

[34] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[35] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distillation using information flow modeling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2339–2348, 2020.

[36] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550, 2014.

[37] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from neighbors: A novel image-level action recognition method. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 39–48, 2017.

[38] Carla Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating actions from a single video frame. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4133–4141, 2017.

[39] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Heterogeneous knowledge distillation using information flow modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2339–2348, 2020.

[40] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating actions in egocentric videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 98–106, 2016.
[41] Vinh Tran, Yang Wang, and Minh Hoai. Back to the future: Knowledge distillation for human action anticipation. arXiv preprint arXiv:1904.04868, 2019.

[42] Basura Fernando and Samitha Herath. Anticipating human actions by correlating past with the future with Jaccard similarity measures. CVPR, 2021.

[43] Xionghui Wang, Jian Fang Hu, Jian Huang Lai, Jianguo Zhang, and Wei Shi Zheng. Progressive teacher-student learning for early action prediction. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019.

[44] Yin Li, Miao Liu, and Jame Rehg. In the eye of the beholder: Gaze and actions in first person video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[45] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 6450–6459, 2018.

[46] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In European conference on computer vision, pages 20–36. Springer, 2016.

[47] F.A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: continual prediction with lstm. In 1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470), volume 2, pages 850–855 vol.2, 1999.