Action Anticipation using Pairwise Human-Object Interactions and Transformers

Debaditya Roy and Basura Fernando

***Abstract*—The ability to anticipate future actions of humans is useful in application areas such as automated driving, robot-assisted manufacturing, and smart homes. These applications require representing and anticipating human actions involving the use of objects. Existing methods that use human-object interactions for anticipation require object affordance labels for every relevant object in the scene that match the ongoing action. Hence, we propose to represent every pairwise human-object (HO) interaction using only their visual features. Next, we use cross-correlation to capture the second-order statistics across human-object pairs in a frame. Cross-correlation produces a holistic representation of the frame that can also handle a variable number of human-object pairs in every frame of the observation period. We show that cross-correlation based frame representation is more suited for action anticipation than attention-based and other second-order approaches. Furthermore, we observe that using a transformer model for temporal aggregation of frame-wise HO representations results in better action anticipation than other temporal networks. So, we propose two approaches for constructing an end-to-end trainable multi-modal transformer (MM-Transformer)[1] model that combines the evidence across spatio-temporal, motion, and HO representations. We show the performance of MM-Transformer on procedural datasets like 50 Salads and Breakfast, and an unscripted dataset like EPIC-KITCHENS55. Finally, we demonstrate that the combination of human-object representation and MM-Transformers is effective even for long-term anticipation.**

I. Introduction

Action anticipation is defined as the task of predicting the occurrence of an action before it starts [16], [41]. There are many applications of action anticipation, e.g., robots assisting humans by predicting upcoming actions [32], autonomous vehicles anticipating pedestrian actions [48], and systems that alert the user if the anticipated action deviates from the correct sequence of actions [54]. It is beneficial to predict future actions before they start to help the control (or decision) systems in these applications. Furthermore, we focus on anticipating those actions that involve the use of objects because the aforementioned applications mostly comprise of such actions.

An effective representation of human-object interactions has been shown to be essential for recognition [46] and anticipation of actions [31] involving objects. Some existing works exploit object affordance (actions possible with an object) and its correlation with the ongoing action label [27]. Other works consider the change in proximity between humans and the various objects throughout an action [46]. An assumption in these approaches is that the number of objects detected in the scene is constant throughout the action. However, objects being interacted with can become occluded [33], or they may appear or disappear from the view due to ego-motion [9]. Another assumption is that all possible object affordances are known beforehand, which may not be possible for all objects in the scene. Hence, we propose a representation to explicitly model pairwise human-object interaction using visual features of humans and objects, and therefore this approach does not require object affordance labels. Specifically, we use cross-correlation across all the pairwise interactions to handle variable number of objects in every frame. Cross-correlation produces a holistic frame representation for the ongoing action that can be used to anticipate future actions.

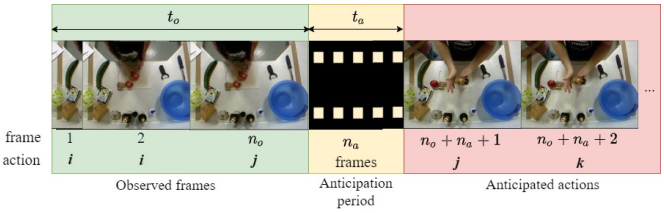


Fig. 1. Frame-wise action anticipation. We observe t_o seconds (n_o frames) of a video and predict actions for future frames after a gap of t_a seconds.

Human action anticipation models can exploit the fact that humans act in a temporally coherent manner where certain actions are always executed in chronological order. For example, during the process of making a salad, "cutting vegetables" generally precedes "seasoning the salad", as shown in Figure 1, where the action of "cutting tomato" during the observation period is followed by the action of "adding pepper" during anticipation. Hence, there is a direct correlation between the actions in the observed frames and future frames. So, action anticipation can be considered as a sequence to sequence (seq2seq) modeling task [57] where the future actions can be generated based on the observed actions. Existing works have used observed action labels to anticipate future action labels using both convolutional and recurrent networks. Convolutional networks can only aggregate information from a local temporal neighborhood dictated by the convolution's kernel size. Similarly, recurrent networks are not adept at learning long-range dependencies where the forward and backward signals have to traverse long paths [25]. Owing to the limitation of convolutional and recurrent architectures, we propose to use Transformers that have been shown to be the most effective on seq2seq modeling tasks [60].

A significant difference between seq2seq modeling in ma-

D. Roy and B. Fernando are with Institute of High-Performance Computing (IHPC), A*STAR, 1 Fusionopolis Way #08-10 Connexis North, 138632, Singapore.

[1]Code at https://github.com/debadityaroy/MM-Transformer_ActAnt

chine translation and action anticipation is that we do not have access to the observed sequence's ground truth actions. Therefore, we rely on encoder-decoder networks [12], [60], [69] that can generate features of future frames from observed frames. We explore different encoder and decoder frameworks in conjunction with transformers to aggregate human-object representations of observed frames and generate features for future frames. We show that transformers perform better temporal aggregation as an encoder and anticipation as a decoder than other encoder-decoder networks [12], [69].

Some of the pioneering works in action anticipation have combined motion and spatial representations with object features to improve the anticipation of future actions [16]. We propose a multi-modal transformer network that aggregates evidence across different representations to generate future frame features. Our end-to-end trainable multi-modal transformer combines the proposed human-object interaction with motion and spatio-temporal features and anticipates actions better for both the immediate and long-term future. Our experiments are performed on two procedural cooking datasets-Breakfast [33], 50Salads [55] where the action sequences are regularly ordered and one unscripted regular activities dataset - EPIC-KITCHENS55 [9] where the actions can be of somewhat arbitrary order.

In summary, our contributions are as follows:

- Representing pairwise human-object interactions with visual features and using cross-correlation to obtain a holistic representation for variable number of human-object pairs in every frame of the observation period.
- Propose a new multi-modal transformer that combines human-object, spatio-temporal, and motion representations to anticipate future actions.
- Robust long-term action anticipation using human-object representation combined with other representations using the multi-modal transformer.

Next, we discuss the related work.

II. Related Work

We discuss three types of related work; human-object interaction methods (section II-A), models for action anticipation (section II-B) and finally methods that use second-order statistics from features for various tasks (section II-B).

A. Human-object interaction

Human-object interactions can provide a deeper understanding of human actions based on the object's affordance and its relative proximity to the human. Understanding and representing human-object interactions has been shown to be effective on a variety of tasks like action specific image retrieval [47], caption generation [70], and question answering [39], [70]. Human-object interactions were used to predict both object and actions using a jointly trained object and action recognition 2D CNNs [28]. In [21], a third input stream called the interaction branch was added to utilize the output from both objects and humans to improve action recognition performance. A fusion technique was proposed in [40] that encoded features from actors, objects, and their spatial relations into a single representation to model actions for zero-shot learning. In [46], the combined feature representation was replaced by a graph-based representation of humans, objects, and their interactions annotated by object affordances. Each node in the graph used a convolutional LSTM [68] to model the evolution of the graph over time for action localization in videos. The convolutional LSTM model used CNNs in every frame for object detection that requires object annotations and affordances for training. In [27], all nodes and edges in the human-object interaction graph were represented by Recurrent Neural Networks to form a structural RNN (S-RNN). Object affordance and activity labels were predicted at each object and human node, respectively. We utilize only visual features to represent human-object interactions as object affordances are challenging to obtain for all types of objects.

In [5], a relation network composed only of objects was proposed to model their temporal evolution for action recognition. The method relied on annotated objects but did not model the relationship between the objects and the actors. An actor-centric network was proposed in [56] to implicitly model the interactions between actors and objects without object annotations for training. Particularly, relational networks avoid explicitly modeling objects by treating each location in an image as an object proxy and aggregating the representations across all the locations. In [71], relevant objects and humans were tracked over time to extract long-term motion patterns called tubelets. The interactions between these tubelets was represented using graph convolution networks for action recognition. All the temporal networks rely on robust tracking of humans and/or objects over time that is affected when objects are occluded [33], objects move in and out of the scene due to ego motion [9], or the human is not visible in some frames [55].

B. Action Anticipation

One of the earliest works presented action anticipation as that task of generating the visual representation of future frames by leveraging the temporal structure of videos [62]. From a single input frame, multiple possible future frames were generated using a regression-based CNN network and subsequently classified to predict the action label. In [41], instead of generating frames, the action label of a frame 1 second into the future was predicted after observing a set number of frames in the recent past. The representation for each observed frame was extracted using a CNN network and sent to a linear model called the predictive model to formulate the sequence's representation. A low-rank linear model called the transitional model was used to predict the future frame label using the sequence representation.

A method that correlates the past video representations with the future for action anticipation using new Jaccard vector similarity measure is presented in [13]. Predicting a sequence of future actions instead of only one future frame was considered in [1], [29], [43]. An RNN and a CNN network were constructed to predict the future action label sequence using only the action labels as input in [1]. On the other

hand, attended temporal features and time-conditioned skip connections were used for anticipating future actions in [29]. A weakly supervised encoder-decoder model is presented in [43]. All these methods [1], [29], [43] observe multiple actions in the past to predict multiple actions in the future. In this paper, we focus on predicting the action labels of multiple frames in the immediate future. Our method also differs from approaches like [18], [49]–[51], [64] that predict future visual representations from a few early frames to predict the ongoing action which is termed as early action recognition. Especially in [64], the problem of early action recognition is solved in a multi-camera setup by integrating information from multiple cameras. The integration is performed using a modified recurrent network that can that can perform action recognition by reconstructing missing information arising due to variable frame rates of different cameras. Different from early action recognition, we consider the case where the labels predicted for the future frames correspond to a different action than the one in the observed frames.

Along with the action labels for each frame, the spatial representation of a frame was added to forecast future action labels in [17]. A neural memory network was proposed that stores information in an LSTM cell by comparing the similarity of the input with the existing memory content. The temporal information for both the labels and spatial streams were propagated using the neural memory networks to obtain better forecasting accuracy. Another approach that considered three frame-based representations - spatial, motion, and object, to predict future actions was proposed in [16]. Using an unrolling LSTM, the authors showed that multiple time-steps in the future could be predicted. A multi-modal attention network was used to decide the best possible combination of the spatial, motion and object representations. Hence, we also propose a multi-modal network to leverage different representations of the observed frames. Particularly, we use transformers to model the temporal attention across each modality inspired by action recognition models using transformers [19], [20]. Transformer based approaches use self-attention to summarize each actor's movement for group activity recognition [19] or the entire scene for simultaneous action detection and localization [20]. In both [20] and [19], transformers are used to process the output of pose networks or I3D spatio-temporal features. Instead, we use transformers for temporal aggregation of frame-based cross-correlated human-object pairs and to summarize features of observed frames as well as generating features of future frames.

Other temporal networks that have been used extensively used for action anticipation include Recurrent Neural Networks (RNN) [7], [44], [48], [53], [58] and Conditional Random Fields (CRF) [31], [67]. The transition matrix in CRF learns all action-to-action transition probabilities that can help in anticipating future actions that are far apart [72]. An RNN like Gated Recurrent Unit (GRU) can learn to represent the entire frame sequence using a single representation that can also be used to predict one or more future action labels (by unfolding) [16]. A graph-based CRF was used to model human-object-action relations for action anticipation in [31] using object affordance labels and action labels for predicting the entire graph for future frames. In [42], different spatial zones were manually identified from egocentric videos along with the actions supported by these zones to form a topological map. Each node in the topological map is populated with visually similar zones from different videos and the nodes are connected sequentially based on the order in which they are visited. For action anticipation, the observed video segment is matched to a node, and the actions corresponding to the next node in the topological map are the anticipated actions. A simpler approach for aggregating both recent and long-term temporal history using non-local blocks [66] for action anticipation was presented in [52]. They showed that while long-term aggregation can sometimes play a part in anticipation, recent actions are more informative in determining the immediate future. Our proposed model that only uses the recent past demonstrates that very limited temporal history is sufficient to accurately anticipate actions not only in the immediate but distant future as well.

C. Second-order statistics from features

Cross-correlation and covariance are both second-order statistics derived from features. Covariance matrices of features have been used extensively for pedestrian detection [59],fine grained image recognition [11], [37], [65], scene categorization [65], and facial expression recognition [2]. Covariance matrices are symmetrical positive definite (SPD) and hence, both Riemannian [45], [59] and Log-Euclidean metrics [3], [8] have been used to compare the similarities across them. With the advent of deep learning, many works have proposed the extraction of second-order statistics from CNN features [26], [37], [65] using covariance pooling. The main difference between the various approaches on covariance pooling in CNNs is the way of normalizing the SPD matrices. Different from the approaches discussed above, we aim to capture the similarity between human-object interactions by their second-order statistics captured using cross-correlation. A cross-correlation matrix is a rectangular matrix that does not possess the SPD properties of the covariance matrix.

Covariance pooling approaches compute the second-order statistics between different representations of the same image either using hand-crafted local features [11], [59] or CNN feature maps [26], [37], [65]. Instead, we use only the detected humans and objects to extract second-order statistics as the focus is human-object interactions. So, the proposed cross-correlation features use a refined subset of the inputs used for covariance pooling. Also, co-occurrence [30] and correlation [63] between a human and a particular object in the frame have been proposed to represent human-object interaction. We were inspired by these ideas to consider correlation between humans and objects as a frame representation. However, humans can possibly interact with multiple objects in the frame during the actions in 50Salads, Breakfast, and EPIC-KITCHENS55 datasets. So, we decided to derive second-order characteristics using cross-correlation to account for all possible human-object interactions.

III. Proposed Approach

In this section, the proposed approach is described in detail. We present the problem statement and an overview of the entire approach. Then, we discuss how cross-correlated human-object features are computed and used with a transformer architecture to generate sequence of action labels. Finally, we describe how a multi-modal transformer is used to combine different representations with human-object features.

A. Problem Statement

We observe a video segment $\mathbf{V}_o = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{n_o}\}$ for t_o seconds containing n_o frames called the *observation period*. The task of action anticipation is to predict a sequence of actions t_a seconds (or n_a frames) after the observation period as shown in Figure 1. In our formulation, the sequence of predicted actions has the same length as the number of observed frames n_o. Hence, the predicted action sequence can be denoted as $\{y_{(n_o+n_a)+1}, y_{(n_o+n_a)+2}, \cdots, y_{(n_o+n_a)+n_o}\}$. A model Φ can be trained to learn a set of parameters Θ that can predict the future action sequence as follows:

$$\{y_{(n_o+n_a)+1}, y_{(n_o+n_a)+2}, \cdots, y_{(n_o+n_a)+n_o}\} = \Phi(\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{n_o}\}, \Theta). \quad (1)$$

Our formulation of the action anticipation task is similar to the sequence to sequence modeling task in machine translation [57]. During training, the model accepts an input sequence that is the observed frame sequence and produces an output sequence, which predicts future action sequence. During testing, we use majority voting on the predicted action sequence to obtain a single anticipated action. It is necessary to obtain a single anticipated action for a fair comparison with existing approaches that predict only a single action from the observed frame sequence.

B. Overview of proposed approach

We use cross-correlation between human and object features to obtain a frame-based representation for every frame in the observation period. Next, we use the sequence of frame-wise human-object (HO) features as input to a transformer encoder and use a transformer decoder to output a sequence of action labels for the immediate future. Finally, we train a multi-modal network that has separate transformer encoders for human-object, spatio-temporal, and motion features. The decoder for the multimodal networks combines the evidence across the various modalities to output a sequence of future action labels.

C. Pairwise human-object interactions

A majority of daily human actions comprise of interacting with various objects. Identifying such actions depends on how well pairwise human-object interactions are represented. In this work, we consider the pairwise relationship between every object and human in a given frame. Figure 2 shows the construction of the frame representation using pairwise human-object interactions. Human(s) and object(s) are detected in every frame using an object detector, and the corresponding regions are cropped from the frame. The cropped regions are used to train a CNN-based classifier to recognize object classes that human interacts. Then a feature representation is constructed from each human and object region in every frame using corresponding human and object features. Specifically, we make use of cross-correlation between human and object features to build the frame representation. Let $\mathcal{P}$ represent all human-object pairs in a frame. Every pair is d_f-dimensional representation, constructed by concatenating a human and an object feature, $\mathbf{h}_{feat}$ and $\mathbf{o}_{feat}$, respectively. As a baseline model, we use a sum of these pairwise features with a shared weight matrix to represent a frame as follows:

$$\mathbf{v} = \sum_{p \in \mathcal{P}} ReLU(\mathbf{W}[\mathbf{h}_{feat}, \mathbf{o}_{feat}]_p^T)), \quad (2)$$

where $[\cdot,\cdot]$ represents vector concatenation, and $\mathbf{W}$ is a learnable projection matrix of size $d_l \times d_f$ that projects the pairwise feature $[h_{feat}, o_{feat}]$ into a lower dimension d_l. We can model $\mathbf{W}$ as a linear layer in a neural network that obtains a compressed representation of the pairwise feature.

The ReLU function used here to provide non-linearity has been shown to be faster and effective for the training of neural networks due to sparse activation, better gradient propagation, simpler operations, and scale invariance [22]. We call this representation as the non-linear projected sum of pairwise features (NPS).

D. Cross-correlation of pairwise human-object features

Different human-object pairs in a frame are important in identifying different actions [46]. Determining the most relevant human-object pair is challenging without the action label being provided. We can only observe the frames in the observation period without the corresponding action labels as per the protocol for action anticipation followed in literature [16], [52]. Hence, we use cross-correlation to learn a holistic representation for all human-object pairs in a frame. At first, we obtain non-linear low-dimensional projections of for each human-object feature as

$$\mathbf{w}_p = (tanh(\mathbf{W}[\mathbf{h}_{feat}, \mathbf{o}_{feat}]_p^T)) \quad (3)$$

where $\mathbf{W}$ is the same learnable projection matrix as in Equation 2. The $tanh$ function is used to associate a negative or positive value to each dimension in the $d_l \times 1$-dimensional weight vector. Then, we use cross-correlation to measure the similarity of every dimension in the pairwise feature to their corresponding weight vectors for all pairs

$$\mathbf{C} = \sum_{p \in \mathcal{P}} \mathbf{w}_p [\mathbf{h}_{feat}, \mathbf{o}_{feat}]_p. \quad (4)$$

The resultant cross-correlation matrix $\mathbf{C}$ is a $d_l \times d_f$ high-dimensional representation for a frame that captures the variability across the weights and the pairwise features across all the pairs. As the number of human-object pairs is not constant over the frames in the observation period, cross-correlation allows us to obtain a fixed-sized representation $\mathbf{C}$ for every frame. Using covariance pooling [11], [37], [65] across concatenated human-object features results in even

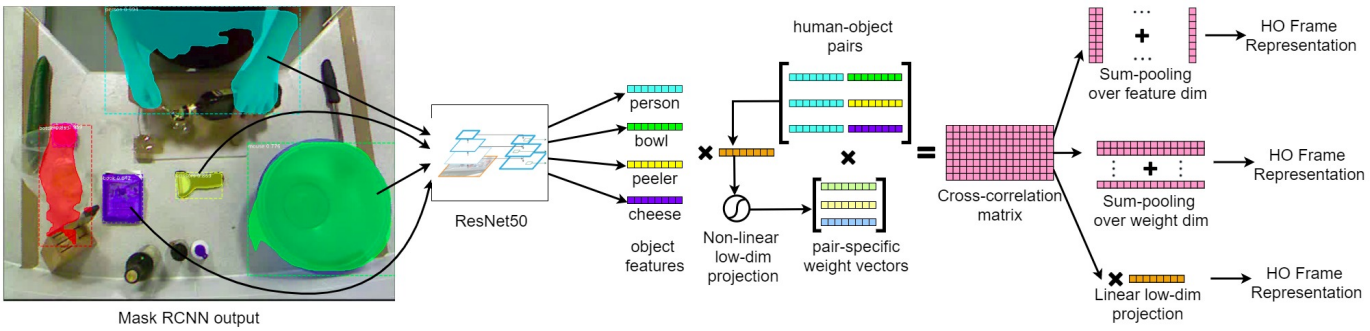


Fig. 2. Constructing a frame representation using cross-correlation of pairwise human-object interactions. Human-object pairwise features are formed with the features of detected objects and human in the frame. The weight vectors of each pair are computed using a non-linear low-dimensional projection. Cross-correlation measures the similarity of every dimension in the pairwise feature to their corresponding weight vectors cross-correlation matrix. Finally, three different frame representations can be obtained from the cross-correlation matrix by a) sum-pooling over pairwise feature dimension, b) sum-pooling over weight vector dimension, or c) linear low-dimension projection.

higher dimensional covariance matrix of size $d_f \times d_f$. Using a high-dimensional cross-correlation or covariance matrix as a frame representation would require estimating a large number of parameters and may lead to overfitting [14]. A method suggested in [37] is to apply average-pooling on the covariance matrix to obtain a bilinear vector. For our cross-correlation matrix, we use both sum-pooling and non-linear projection to encode the higher-order statistics captured in the cross-correlation matrix in a low-dimensional representation.

- *CC(F)*- Sum-pooling over pairwise feature dimension d_f to aggregate cross-correlation across the dimensions of the concatenated features. The frame representation is a $d_l \times 1$ vector computed as follows:

$$\mathbf{v} = [v_1, v_2, \cdots, v_i]^T, \qquad v_i = \sum_j c_{ij} \tag{5}$$

  where c_{ij} represents the $(i,j)^{th}$ element of $\mathbf{C}$.
- *CC(W)*- Sum-pooling over weight vector dimension d_l to aggregate cross-correlation across the dimensions of the projected concatenated features. The frame representation is a $d_f \times 1$ vector calculated as follows:

$$\mathbf{v} = [v_1, v_2, \cdots, v_j]^T, \qquad v_j = \sum_i c_{ij}. \tag{6}$$

- *CC(LP)*- Low-dimensional projection using a linear layer $\mathbf{w}$ of dimension $d_l \times 1$ to obtain a frame representation of dimension $d_f \times 1$ as follows:

$$\mathbf{v} = \mathbf{w}^T \sum_{p \in \mathcal{P}} \mathbf{w}_p [\mathbf{h}_{feat}, \mathbf{o}_{feat}]_p^T. \tag{7}$$

While performing a particular action, the object(s) being interacted with are closer to the human compared to other objects in the scene. So, we concatenate the displacement between humans and objects to the individual weight vectors computed in Equation 3 as follows

$$\mathbf{w}_p^d = tanh(\mathbf{W}_d[\mathbf{d}_p, \mathbf{w}_p^T]^T) \tag{8}$$

where $\mathbf{W}_d$ is a $d_{ll} \times d_l + 2$ matrix that projects the concatenated weight vector from $(d_l + 2)$ dimensions (+2 for the displacement) into a lower dimension d_{ll}. The term $\mathbf{d}_p$ is the normalized displacement vector (with respect to the frame size) between center of the human and center of the object in the pair p, as detected in the frame by the Mask R-CNN architecture [24]. The cross-correlation matrix can then be determined similar to Equation 4 as

$$\mathbf{C}^d = \sum_{p \in \mathcal{P}} \mathbf{w}_p^d [\mathbf{h}_{feat}, \mathbf{o}_{feat}]_p). \tag{9}$$

We can obtain different frame representations (CC-D) from the displacement-based cross-correlation matrix using the techniques described above for the regular cross-correlation matrix.

E. Multi-modal Transformers

The set of cross-correlation based frame representations obtained for the observation period $\mathbf{V}_o = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{n_o}\}$ are used to predict the future action sequence. Our approach consists of comparing the similarity of frame features in the observation period using self-attention. Then, we use these self-attention scores along with the cross-correlation based frame representations to predict the labels for the future sequence. Self-attention has been shown to very effective for computing similarities across tokens in a sequence to perform sequence to sequence translation tasks [61]. Instead of directly comparing the cross-correlation based frame features, we use three abstractions called query, key, and value obtained from the frame features using learnable weights $\mathbf{W}^q, \mathbf{W}^k$, and $\mathbf{W}^v$, respectively, as

$$\mathbf{Q} = \mathbf{V}_o \mathbf{W}^q, \mathbf{K} = \mathbf{V}_o \mathbf{W}^k, \text{ and } \boldsymbol{\Upsilon} = \mathbf{V}_o \mathbf{W}^v. \tag{10}$$

The matrices $\mathbf{Q}$, $\mathbf{K}$, and $\boldsymbol{\Upsilon}$ represent the packed query, key, and value vectors for each frame in the observed sequence. The self attention score vectors for the human-object representation $\mathbf{A}_{HO} = [\mathbf{a}_1^T, \cdots, \mathbf{a}_{n_o}^T]^T$ are based on the similarity of the query and key vectors across the observation sequence scaled by $\sqrt{d_k}$, i.e., the dimension of the key vector as

$$\mathbf{A}_{HO} = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\boldsymbol{\Upsilon}. \tag{11}$$

The entire encoder-decoder process in the transformer model is shown in Figure 3. Finally, a classifier is applied on the anticipated feature sequence to obtain the action labels.

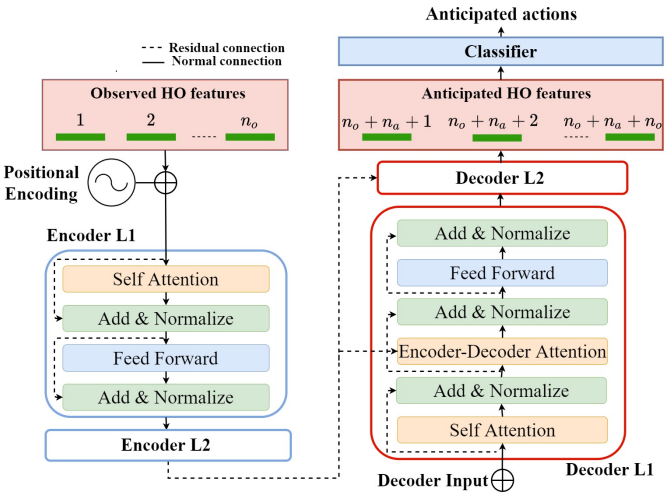


Fig. 3. Anticipation of future action labels using the encoder-decoder transformer model.

Using multiple representations of the same video sequence has been shown to improve anticipation performance [16]. So, we propose to utilize the spatial-temporal, motion, and the proposed human-object interaction-based representation from each frame to exploit the complementary information in each of them. While the spatial-temporal (ST) representation provides how the entire scene changes while the action is being performed, the motion representation describes only those parts of the video that move during each action. Finally, the human-object interaction captures the relationship between the actor and the objects.

We propose two mechanisms for combining these representations - i) separate encoders for each modality with a shared decoder as shown in Figure 4(a), and ii) separate encoder-decoder for each modality with pooled output as shown in Figure 4(b). Using a shared decoder for multiple encoders is inspired by the multi-source language translation task that takes in multiple input sequences to produce a single output sequence [35]. The attention layer in the encoder computes self-attention independently over each input modality. The resulting contexts are then treated as states of an input and self-attention is computed once again as follows

$$\mathbf{K}_{multi} = \boldsymbol{\Upsilon}_{multi} = concat(\mathbf{A}_{HO}, \mathbf{A}_{ST}, \mathbf{A}_{M})$$
$$\mathbf{A}_i = softmax(\frac{\mathbf{Q}\mathbf{K}_{multi}}{\sqrt{d_k}})\boldsymbol{\Upsilon}_{multi}, i \in \{HO, ST, M\} \quad (12)$$

The entire multi-modal network is fine-tuned to learn the weights of the decoder. The concatenation of the encoder states is performed after observing the entire frame feature sequence instead of every frame as in Modality Attention [16].

Another way to combine the evidence across modalities is to pool the classifier outputs of the individual transformers. The classifier produces class-wise confidence scores for every anticipated feature generated by the decoder. Confidence scores are accumulated across the classifiers from each modality and added for every output class separately. We repeat this accumulation and addition process for every anticipated feature. The final predicted action for each anticipated feature is based on these consolidated confidence scores computed across the modalities. The entire multi-modal network is trained using the cross-entropy loss between the target actions and the final predicted actions.

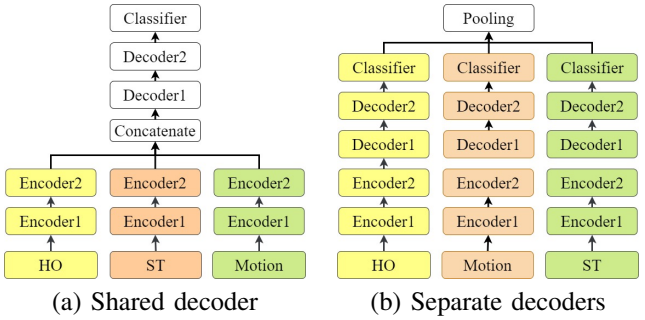


Fig. 4. Proposed mechanisms for designing multi-modal transformers.

IV. Experiments and Results

In this section, we describe various experiments for action anticipation on 50 Salads, Breakfast, and EPIC-KITCHENS55.

A. Datasets and Features

50 Salads [55] dataset consists of 50 videos of 25 actors making salads based on recipes provided beforehand. The videos are recorded with a resolution of 640×480 at 30 frames per second. The actors perform 17 different fine-grained actions, and the gaps between these actions are annotated using a background class. The average video length is 6.4 minutes, and there are 20 action instances per video. The published dataset provides five splits, and all the results presented here are averaged over the five splits.

Breakfast [33] dataset consists of 77 hours of procedural videos or 4.1 million frames of 52 actors making breakfast that yields 48 fine-grained action classes. Indeed, it is a large scale video dataset. The videos are recorded with a resolution of 320×240 at 15 frames per second. The average duration of the videos is comparably shorter at 2.3 minutes with an average of 6 action instances. All the results presented here are averaged over the four splits provided by the authors of the dataset [33].

EPIC-KITCHENS55 [9] contains 55 hours of unscripted videos comprising 39,596 action annotations, 125 verbs, 351 nouns, and 2,513 actions. All the videos are recorded at 60 frames per second with a resolution of 1920×1080. We downsample the videos to 10 frames per second for our experiments. The training set is divided into 232 videos for training (23,493 segments), and 40 videos for validation (4,979 segments) based on the splits provided by [16]. We also evaluate our results using the test set of EPIC-KITCHENS55.

The 50Salads dataset has top-down videos, and this viewpoint is not encountered often in the COCO dataset [36] that is used for training the Mask R-CNN architecture. Hence, we manually segregate the cropped images of each detected object into 10 categories - bottle, bowl, cheese, cucumber, knife, lettuce, peeler, spoon, tomato, and hands (human) based on the actions in the dataset. Then, we train a classifier based on

TABLE I
COMPARISON OF CROSS-CORRELATION USING DIFFERENT OBJECT DETECTORS ON ACTION ANTICIPATION

Detector	50 Salads	Breakfast
SSD [38]	38.5	29.6
Mask R-CNN [24]	**41.7**	**31.6**

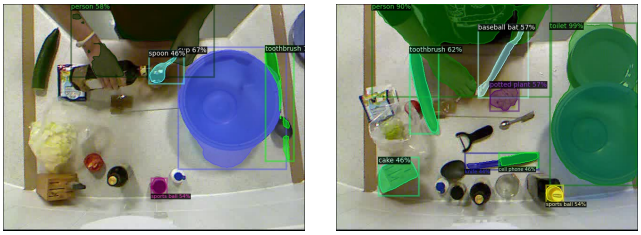


Fig. 5. Examples of Mask R-CNN detections from the 50 Salads dataset. Humans and objects being interacted with are detected though the class labels for objects are not correct. The detection outputs are segregated manually by the correct object type and a classifier based on ResNet50-FPN [24] architecture is learnt to be used during inference.

TABLE II
COMPARISON OF DIFFERENT CROSS-CORRELATION METHODS ON ACTION ANTICIPATION

Cross-correlation type	Observation period			
	1s	2s	3s	5s
50 Salads				
NPS	*37.3*	36.2	36.1	35.1
CC(F)	***41.7***	38.9	38.4	37.5
CC(W)	*41.5*	38.8	38.3	37.4
CC(LP)	*41.4*	38.3	38.1	37.2
CC-D(F)	*41.1*	39.1	38.3	37.4
CC-D(W)	*41.1*	39.0	38.2	37.3
CC-D(LP)	*40.9*	39.1	38.1	37.1
Breakfast				
NPS	*29.4*	28.2	27.3	27.4
CC(F)	***31.6***	30.4	29.6	29.2
CC(W)	*31.4*	30.3	29.5	29.3
CC(LP)	*31.3*	30.2	29.7	29.4
CCD(F)	*31.3*	30.9	29.1	28.5
CCD(W)	*31.2*	30.8	29.3	28.4
CCD(LP)	*31.1*	30.9	29.1	28.6

the ResNet-50 Feature Pyramid Network (FPN) [24] for these 10 categories. The trained classifier achieves a mean accuracy of 97.5% which means that both objects and hands are detected with high confidence. Further, we can use the features from the penultimate layer of this classifier to get distinct visual features for different objects and humans. Hence, during testing, we run the Mask R-CNN on every frame and extract 256-dimensional features from each detected object. The features are then used for two tasks - a) classification into one of the 10 categories mentioned above, and b) forming the human-object pairs.

The Mask R-CNN network [24] object detections per frame is shown in Figure 5 for 50Salads. For the Breakfast dataset, we consider 15 categories of common kitchen objects plus the person category from the COCO dataset. As the Mask R-CNN is already trained on the COCO dataset, we can directly extract features from the ResNet-50 FPN network. A trained Faster R-CNN based object detector and hand masks are provided for the EPIC-KITCHENS55 dataset [9]. The detected objects and hand masks can be directly used to obtain the features as the other datasets.

Detecting objects accurately in every frame is vital for the current proposed framework based on human-object interaction. To demonstrate the effect of Mask R-CNN we replace it with a weaker detector like SSD [38] based on the performance on COCO dataset). Table I shows that using a stronger detector like Mask R-CNN improves the action anticipation performance compared to SSD for both 50 Salads and Breakfast dataset. Therefore, we resort to stronger object detector such as Mask R-CNN in the rest of the experiments.

B. Overview of Experiments

In the following subsections, the results of many ablation studies are presented. In Section IV-C, we compare the proposed cross-correlation based features with self-attention and covariance pooling. Section IV-D compares the transformers to various temporal networks both as an end-to-end architecture and separately as an encoder or a decoder. Finally, in Section IV-D, we show the effect of incorporating different features in the multi-modal transformer and then compare with state-of-the-art approaches on three different datasets in Section IV-F and IV-G.

C. Comparison of Human-object representations

For both the 50 Salads and Breakfast dataset, we consider an anticipation period of 1 second for a fair comparison with existing approaches [1], [41], [62]. The transformer encoder-decoder architecture consists of 2 encoders and 2 decoders. Every encoder and decoder has a hidden layer dimension of 64, feed-forward layer dimension of 2048, and 2 attention heads based on empirical performance. To determine the optimal observation period, we ran an ablation study with observation periods of 1, 2, 3, and 5 seconds to denote varying amounts of recent temporal history also used in [52].

Using the transformer architecture described above, we compare the performance of different human-object representations (discussed in Section III-C) in Table II. All three techniques for obtaining low-dimensional representation from the cross-correlation matrix CC(F), CC(W), and CC(LP) produce similar performance, which shows that the projection method is not crucial for frame representation. Finally, adding displacement to the cross-correlation (CC-D) does not improve anticipation performance, indicating that displacement does not contribute to refining the choice of the most important human-object interaction in the frame. We use the best performing CC(F) and CC-D(F) versions of the CC and CC-D representations in the rest of the experiments.

We compare cross-correlation to other approaches that compare similarities across features like scalar attention weights [4], self-attention [61] and covariance pooling [37]. For covariance pooling, we compute the covariance matrices between i) concatenated human-object features (concat), and ii) human feature with every object feature in the frame (paired). Then, we perform i) average-pooling following [37] to produce a fixed-dimensional frame representation called

TABLE III
COMPARISON OF CROSS-CORRELATION WITH OTHER COMBINATION METHODS ON ACTION ANTICIPATION

Human-Object (HO) Representation	Observation period			
	1s	2s	3s	5s
50 Salads				
Vec. cov. (concat)	*37.4*	35.2	36.5	34.1
Vec. cov. (paired)	*37.1*	35.4	36.2	34.2
SVD cov. (concat)	37.3	35.8	36.5	35.7
SVD cov. (paired)	*37.8*	35.7	36.2	36.1
Bilinear cov. (concat) [37]	*39.8*	39.2	38.2	37.1
Bilinear cov. (paired) [37]	*39.6*	38.5	37.5	36.2
Attention [4]	*37.6*	36.5	36.2	35.9
Self-Attention [60]	*38.2*	36.9	36.6	36.1
Self-Attention (obj)	*38.2*	36.9	36.6	36.1
Cross-correlation	***41.7***	38.9	38.4	37.5
Breakfast				
SVD cov. (concat)	*29.5*	28.7	28.5	28.2
SVD cov. (paired)	*29.4*	28.8	28.6	28.1
Vec. cov. (concat)	28.5	26.7	28.5	27.2
Vec. cov. (paired)	*28.4*	27.8	27.6	24.7
Bilinear cov. (concat) [37]	*29.9*	29.7	27.5	27.4
Bilinear cov. (paired) [37]	*29.4*	29.6	29.2	27.3
Attention	*29.2*	28.1	27.1	27.3
Self-Attention	*30.2*	28.8	28.2	27.9
Self-Attention (obj)	*30.6*	28.7	28.1	27.8
Cross-correlation	***31.6***	30.4	29.6	29.2

bilinear vector in [37], ii) vectorizing the covariance matrix and applying a linear projection to obtain a low-dimensional representation called vectorized covariance vector (Vec. Cov.), and iii) singular value decomposition of covariance matrix to obtain singular value matrix that is vectorized to get a singular value vector (SVD Cov.). The bilinear vector, vectorized covariance, or singular value vector, is used to train the transformer similar to the sum-pooled cross-correlation frame representation CC(F). Finally, we also compare against a modified self-attention network that computes the similarity between the human feature (query) and all the object features (keys) in the frame. Each concatenated human-object feature (value) is then weighted by its corresponding similarity score normalized across all pairs and denoted as Self-Attention (obj) in Table III.

Table III shows that cross-correlation performs better than scalar attention, self-attention, and covariance pooling for both datasets. Among covariance pooling methods, bilinear pooling demonstrates the best anticipation performance and is even better than attention-based methods. Comparing second-order methods like covariance and cross-correlation across concatenated human-object pairs shows that cross-correlation is more effective for action anticipation. The proposed cross-correlation approach is able to better represent the human-object interactions in a frame compared to covariance pooling.

Further, as shown in Table II and III, the observation period of 1 second is optimal for anticipation accuracy across different human-object representations. Furthermore, for both 50 Salads and Breakfast, the anticipation accuracy deteriorates as the observation period increases. As increasing the observation period leads to longer anticipation sequences, the network must anticipate further into the future that introduces more uncertainty. So, the likelihood of making false predictions increases which affect overall anticipation accuracy.

D. Comparison of Temporal Networks

In this subsection, we evaluate the performance of different temporal networks as shown in Table IV and V. The different temporal networks used for comparison with the transformer model are described below.

- *GRU+CRF*- The HO features for every frame in the observation period are passed as input to a single layer GRU with a hidden unit dimension of 32 to add temporal information from the preceding frames. A classifier is used on the hidden state for every frame in the observation period to get the observed action labels. Then, a CRF is trained to learn the transitions between the observed action and anticipated action labels.
- *MS-TCN Encoder-Decoder* [12]- Multi-stage Temporal Convolution Networks is a multi-stage architecture for temporal action segmentation. At each stage, a set of dilated 1D temporal convolutions is used to generate an initial prediction refined by the next stage. In our implementation, we use 4 stages of 1D convolutions in both the encoder and decoder following [12].
- *TempRec Encoder-Decoder* [69]- Temporal Recurrent Networks have an encoder-decoder structure. The encoder is an RNN that has a hidden state output for every frame representation in the observation period. The decoder generates a sequence of future features using each hidden state output. The future features are combined to produce a "future" context according to [69]. The hidden state and future context are concatenated to obtain each observed frame's action label. In our implementation, we use the future context of each observed frame to anticipate future action labels.
- *TempRelNet* [73]- Temporal Relation Networks learn temporal dependencies between video frames at multiple time-scales for action recognition. Each time-scale is represented by the temporal relationship between a particular number of frames. We consider temporal relations comprising of d frames - $d = \{4, 5, 6, 7\}$ for 50 Salads and $d = \{2, 3, 4\}$ for Breakfast. The frame relations are based on the number of observed frames for Breakfast (15) and 50 Salads (30). For each d, we choose 3 random samples from the observation period and aggregate their corresponding HO representations using a linear layer [73]. Finally, the aggregated representations are added to obtain a representation for the observed frame, which is used then used for anticipating a single future action after the anticipation period.

In both Table IV and V, we compare the different temporal networks described above using different human-object representations. As a baseline frame representation, we consider the average of object and human features in every frame (AVG). All temporal networks benefit from using HO representations rather than AVG for anticipation. The GRU+CRF lags behind other temporal networks for all the frame representations. The classifier used to obtain observed action labels from the GRU hidden states can lead to misclassification errors that

TABLE IV
PERFORMANCE OF TEMPORAL NETWORKS ON 50 SALADS

Temporal Network	AVG	HO Representation		
		NPS	CC	CC-D
GRU+CRF	23.8	30.1	26.9	26.2
MS-TCN Encoder-Decoder [12]	26.5	34.3	36.3	36.1
TempRec Encoder-Decoder [69]	28.3	34.6	36.8	35.9
TempRelNet [73]	29.8	36.1	36.9	36.7
Transformer Encoder-Decoder	*33.4*	*38.3*	***41.7***	*41.2*

TABLE V
PERFORMANCE OF TEMPORAL NETWORKS ON BREAKFAST

Temporal Network	AVG	HO Representation		
		NPS	CC	CC-D
GRU+CRF	9.3	13.1	14.2	14.1
MS-TCN Encoder-Decoder [12]	19.3	25.9	27.5	26.2
TempRec Encoder-Decoder [69]	19.6	27.8	28.4	28.3
TempRelNet [73]	21.5	*29.6*	31.2	31.3
Transformer Encoder-Decoder	*22.4*	*29.4*	***31.6***	*31.5*

adversely affect the CRF's anticipation. Interestingly, TempRelNet performs better than MS-TCN and TempRec which demonstrates that considering sequences of various lengths in the observation period is more effective in capturing temporal context than temporal convolution or recurrence. However, the best performance is obtained by the transformer model that employs self-attention for capturing temporal context and anticipating future actions. Also, we found that the using 2 encoder and 2 decoder layers in the transformer network produces the best anticipation accuracy as shown in Table VI.

To understand the efficacy of transformers, it is important to study the influence of the encoder and decoder in anticipation. We measure the performance of the transformer encoder by using different networks as a decoder -

- *linear* layer,
- *CRF* with action labels predicted from the transformer encoder,
- *MS-TCN decoder* with average pooled output from the transformer encoder for every stage (as the decoder expects a single output from the encoder), and
- *TempRec decoder*.

Similarly, we design networks with the transformer model as decoder but use different encoders

- *GRU* hidden states for every observed frame,
- *MS-TCN encoder* with one output per stage (4 as per [12]) that are padded to match the number of frames in the observation period, and
- *TempRelNet* with the same number of sub-samples as the number of frames in the observation period.

TABLE VI
EFFECT OF TRANSFORMER LAYERS ON ACTION ANTICIPATION

Encoder layers	Decoder layers	50Salads	Breakfast
1	1	35.6	24.8
2	1	38.9	29.6
1	2	38.1	29.3
2	2	**41.7**	**31.6**

TABLE VII
PERFORMANCE OF TRANSFORMER ENCODER AND DECODER WITH OTHER TEMPORAL NETWORKS

Encoder	Decoder	Anticipation Accuracy	
		50Salads	Breakfast
Linear	Linear	23.2	9.8
Transformer	Linear	*34.3*	*19.9*
GRU	CRF	26.9	14.2
GRU	*Transformer*	*35.6*	*22.5*
Transformer	CRF	32.9	20.5
MS-TCN	MS-TCN	36.3	28.7
Transformer	MS-TCN	36.4	*29.4*
MS-TCN	*Transformer*	*36.8*	29.1
TempRec	TempRec	36.8	*30.6*
Transformer	TempRec	*37.1*	30.5
TempRelNet	Linear	36.9	29.9
TempRelNet	*Transformer*	*39.2*	*30.9*
Transformer	*Transformer*	***41.7***	***31.6***

In Table VII, the performance of the various networks considered above are presented in relation to their non-transformer based counterparts from Table IV and V. All the experiments are conducted using the cross-correlation based human-object interaction feature (CC). Compared to the GRU+CRF network, adding either the transformer encoder to replace the GRU or the transformer decoder to replace the CRF improves the performance for both 50 Salads and Breakfast dataset. Hence, self-attention is more effective at capturing temporal context than GRU and anticipating actions than a CRF. For both the MS-TCN and TempRec decoder, the transformer encoder output has to be averaged across the entire sequence before anticipation, which negates the use of self-attention, and no significant improvement is observed. As a decoder, the transformer can anticipate better with TempRelNet encoder as input compared to GRU or MS-TCN. Hence, the temporal context captured using sub-samples of different lengths in TempRelNet is more descriptive for anticipation than recurrence or temporal convolution. After comparing the various networks, we can observe that replacing the encoder or decoder with transformers improves the anticipation performance. Hence, we can conclude that self-attention is equally useful in both the encoding and decoding processes.

E. Performance of multi-modal transformers

Providing multiple representations of the frame can benefit action anticipation by adding complementary information to the proposed pairwise human-object representation. We provide spatial-temporal representation in the form of I3D features for both Breakfast and 50 Salads datasets as they have been shown to perform better at action anticipation than similar features like R(2+1)D [52]. Following the protocol in [52], we use the 2048-dimensional frame-wise I3D features for Breakfast provided along with the dataset [33] and the I3D features provided by [12] for the 50 Salads dataset. As I3D features are 2048 dimensions compared to 512-dimensional HO representation, we use larger (512-dimensional) hidden layers in the transformer's encoder and decoder for the I3D features. In addition to spatio-temporal features, we use motion

TABLE VIII
COMPARISON OF MODALITIES AND MULTI-MODAL TRANSFORMERS

Modality	Multi-modal Network	Anticipation Accuracy	
		50 Salads	Breakfast
HO(CC)	-	41.7	31.6
Spatio-Temp. (I3D)	-	39.4	15.8
Motion (Dense Traj.)	-	43.1	17.1
Spatio-Temp. (ST) + Motion (M)	Shared Decoder (MM-Sha)	44.1	19.7
HO + ST		40.2	40.2
HO + Motion		43.7	41.1
HO + ST + M		44.6	43.5
ST + M	Separate Decoders (MM-Sep)	44.5	22.8
HO + ST		41.2	40.6
HO + M		44.1	41.7
HO + ST + M		**46.8**	**44.9**

TABLE IX
EFFECT OF FEATURE, COMBINATION, AND ANTICIPATION NETWORK ON ACTION ANTICIPATION

Feature	Combination Mechanism	Anticipation Network	50Salads	Breakfast
HO	NPS	Linear	21.4	7.9
HO	**CC**	Linear	23.2	9.8
HO	CC	**Transformer**	41.7	31.6
HO+ST+M	CC (for HO)	**Multi-modal Transformer**	**46.8**	**44.9**

information in the form of 64 dimensional Fisher vectors of Dense Trajectory features provided by [34] for the 50 Salads dataset and provided by [33] for the Breakfast dataset.

We compare the two multi-modal transformer architectures, as discussed in Section III-E. The individual transformer encoder-decoders are trained for 10 epochs with a learning rate of 0.001 using the Adam optimizer. Then, the multi-modal network is fine-tuned for 5 epochs with a learning rate of 0.0001. As shown in Table VIII, both spatial-temporal and motion representation are effective at action anticipation. However, the best performance is obtained with the multi-modal transformers involving all three modalities. Interestingly, multi-modal transformers can exploit complementary information in motion and spatial-temporal features for Breakfast dataset, where a substantial improvement is seen compared to the individual modalities. In the 50 Salads dataset, the multi-modal transformer provides only a modest improvement as all modalities have similar anticipation performance. Furthermore, using separate decoders and pooling the outputs shows better performance than using a shared decoder, demonstrating that individual decoders play an important role in anticipation. Even when two input modalities are combined, separate decoders perform slightly better than the shared decoder architecture that shows that a shared decoder may not be able to leverage all the information from the different incoming encoders.

Finally, in Table IX, we summarize the effect of each component in our approach on the overall anticipation accuracy across the two datasets - Breakfast and 50 Salads.

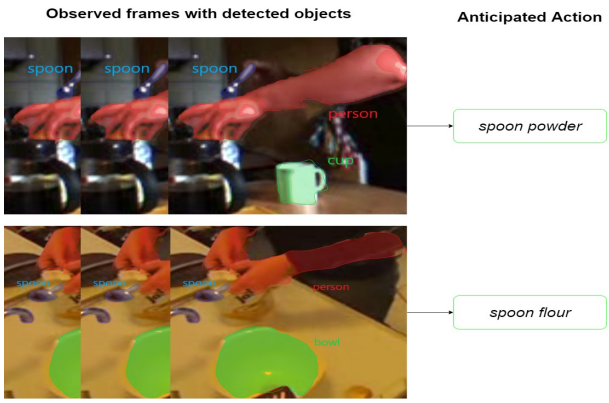


Fig. 6. Examples of objects detected in the observation period leading to the correctly anticipated action. *Spoon powder* is anticipated due to *spoon* and *cup* being detected in the observed frames while *spoon flour* is anticipated based on the detection of *spoon* and *bowl*.

F. Comparison with state-of-the-art:50Salads & Breakfast

In Table X, we compare the performance of the proposed human-object representation and multi-modal transformer on 50Salads dataset. The frame information for a single frame was used for anticipation using a regression network in [62]. Temporal context was added to the frame information obtained in [62] using an RNN. In [1], both an RNN and a CNN were employed to aggregate the action sequence information in the observed frames. Temporal aggregation using non-local blocks was used in [52]. Even with the human-object representation, we were able to outperform all the existing approaches. Hence, accounting for human-object interactions is essential in anticipating actions involving objects. Some qualitative examples are shown in Figure 6.

TABLE X
COMPARISON WITH STATE-OF-THE-ART ON 50 SALADS

Method	Anticipation Accuracy
Deep Regression [62]	8.1
RNN [1]	30.1
CNN [1]	29.8
Temporal Aggregation (w/o segmentation) [52]	40.7
HO(CC) + Transformer	41.7
Multi-modal Transformer (MM-Sep)	**46.8**

In Table XI, we compare our approach with the existing approaches on the Breakfast dataset. Apart from the approaches used in 50Salads, we also compare with a prediction and transitional model proposed in [41]. We can observe that the human-object representation performs not as well as the existing approaches, which contrasts the results on 50Salads dataset. We attribute this to poor object detection in the Breakfast dataset due to occlusion due to many camera views compared to the top-down perspective in 50 Salads, as shown in Figure 7. Further, it is challenging to detect objects in frames of size 320×240 compared to 640×480 in the 50Salads dataset. Combining spatio-temporal and motion representations using the multi-modal transformer leads to an improvement over other methods. Multi-modal transformers

TABLE XI
COMPARISON WITH STATE-OF-THE-ART ON BREAKFAST

Method	Anticipation Accuracy
Deep Regression [62]	6.2
RNN [1]	30.1
CNN [1]	27.0
Predictive+ Transitional [41]	32.3
Temporal Aggregation (w segmentation) [52]	47.0
Temporal Aggregation (w/o segmentation) [52]	40.7
Multi-modal Transformer (MM-Sep) (HO + ST + M)	44.9
MM-Sep (HO + ST + M + frame-wise labels)	**48.4**

(a) (b)

Fig. 7. Examples of scenes with occluded objects in the Breakfast dataset.

with recent temporal context (1 second) fare better than temporal aggregation over longer contexts (5 to 30) seconds [52]. The temporal aggregation framework anticipates better when information about the start and end of action segments is provided. We also incorporated frame-wise ground truth action labels as an added modality to our multi-modal transformer and saw an improvement of 3.5% which outperforms the temporal aggregation approach.

G. Performance comparison on EPIC-KITCHENS55

EPIC-KITCHENS55 is a dataset where the actors perform daily activities in the kitchen without a script. The action anticipation problem consists of predicting the correct verb and noun simultaneously. Hence, we train two different transformer networks for nouns and verbs with an increased hidden layer dimension of 512 in the encoders and decoders due to many noun (125) and verb (351) classes. For a fair comparison with existing methods, we also include results for the Top-5 anticipation accuracy as has been suggested by the authors of EPIC-KITCHENS55 [15]. As RGB and optical flow frames were provided in the dataset, I3D features were extracted on both.

In Table XII (a), we compare the results of existing approaches on the validation set of EPIC-KITCHENS55. The other approaches comprise of the Verb-Noun Cross-Entropy (VN-CE) [15], Temporal Segment Networks combined with SVM Top-5 loss (TSN+SVM) [6], rolling-unrolling LSTM (RU-LSTM) [16], prediction+translation [41]. From these results, we can see that "HO (CC)" alone does not perform well. Even when we make use of spatial-temporal features and motions features, still the performance is somewhat limited.

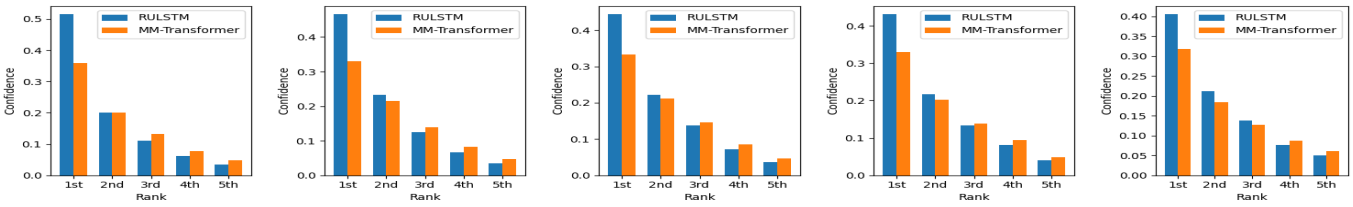
The bag-of-objects (BOO) features are histograms of detected objects in a frame normalized by the total number of appearances of every object in the entire training set. The BOO features can explicitly emphasize which objects are in the frame compared to all objects. Hence, we obtain better anticipation performance when BOO features are added to the multi-modal transformer on the validation set.

We also compare the performance of the proposed method on test-sets of EPIC-KITCHENS55 in Table XII (b) and (c).[2] In terms of noun anticipation, multi-modal transformer outperforms the existing approaches both in terms of top-1 and top-5 accuracy. The multi-modal transformer can generalize well to unknown surroundings and predict the next noun. Multi-modal transformer performs poorly for verb anticipation compared to temporal aggregation [52] or RU-LSTM [16] in-terms of top-1 accuracy. The verb anticipation performance of our model is comparable (on S1) or better (on S2) than existing approaches in-terms of top-5 accuracy. In an unscripted dataset like EPIC-KITCHENS55, multiple future verbs can naturally follow a given observation and top-k accuracy has been proposed as a more natural way to quantify performance [15].

We analyzed the confidence scores of verb anticipation between RU-LSTM and MM-Transformers when the 1st, 2nd, 3rd, 4th, and 5th predictions are correct on EPIC-KTICHENS55 validation set as shown in Figure 8. The analysis was done on the validation set as the test set ground truth is not available. The average confidence score of the top prediction (Rank 1) MM-Transformer is much lower than RU-LSTM while the next prediction confidence scores are almost equal for both the methods. So, there is lower confidence in the correct prediction that increases the chances of incorrect predictions. This can be reason why the top-1 accuracy of MM-Transformer is lower than RU-LSTM and temporal aggregation (uses RU-LSTM) on test sets. Another observation is that the confidence scores for 2nd, 3rd, 4th, and 5th predictions are better than RU-LSTM for all cases (1st, 2nd, 3rd, 4th, or 5th predictions are correct). Hence, it is highly likely that the target verb will be among the top-5 predictions and so, MM-Transformer has comparable or better top-5 accuracy even with lower top-1 accuracy. The property of transformers excelling especially in top-k accuracy over LSTM architectures has also been reported for other prediction tasks like next point of interest recommendation [23].

Somewhat surprisingly, despite our top-5 noun and verb anticipation performance is comparable or better than prior state-of-the-art methods in both S1 and S2 sets, our action anticipation performance remains poor on both S1 and S2. This can be due the fact that when our noun model is correct, it seems the verb model is not and vice-versa. This could lead to poor action anticipation results. Secondly, 92% of the actions in EPIC-KITCHENS dataset have fewer than 5 examples in total across training, validation and two test sets [10]. Consequently, many of the actions may not be observed during training. So, it is possible that we are not able to learn the human-object interactions in many actions during training. When these actions appear in test set, our model is

[2]Results on test server are under the username `debadityaroy`



(a) 1st rank correct (b) 2nd rank correct (c) 3rd rank correct (d) 4th rank correct (e) 5th rank correct

Fig. 8. Comparison of average confidence scores for verb anticipation between RU-LSTM and MM-Transformer when different ranked predictions are correct on EPIC-KITCHENS55 validation set for verb anticipation. MM-Transformer scores are higher than RU-LSTM on average which means the there is a higher chance that will be included in Top-5 predictions. The top prediction confidence is lower than MM-Transformer is lower than RU-LSTM and can lead to lower Top-1 accuracy.

not able to anticipate such combinations of nouns and verbs at the same-time. However, as nouns and verbs are shared by many actions, still the top-5 noun and verb performance is satisfactory or better than state-of-the-art methods.

TABLE XII
COMPARISON WITH STATE-OF-THE-ART ON EPIC-KITCHENS55

Method	Top-1 Anticipation Accuracy			Top-5 Anticipation Accuracy		
	VERB	NOUN	ACT.	VERB	NOUN	ACT.
(a) Validation Set						
VN-CE [9]	31.77	15.81	5.79	77.67	39.50	17.3
TSN+SVM [6]	25.65	15.99	11.09	72.70	38.41	25.42
RU-LSTM [16]	32.66	21.74	14.18	79.55	51.79	35.32
HO (CC) + Transformer	24.57	13.65	9.34	67.62	35.21	21.34
I3D-RGB (ST) + Transformer	23.42	11.32	10.17	65.45	32.19	21.44
I3D-Flow (M) + Transformer	24.51	10.35	9.15	63.54	30.95	20.65
Bag of objects (BOO) + Trans.	27.63	22.14	13.52	73.54	51.95	32.65
MM-Sep (HO+ST+M)	27.51	14.62	12.56	67.22	35.96	21.43
MM-Sep (HO+ST+M+BOO)	**32.93**	**22.91**	**14.62**	**79.54**	**52.07**	**35.65**
(b) Test Set - Seen Kitchens (S1)						
Pred+Trans [41]	30.70	16.50	09.70	76.20	42.70	25.40
TSN+SVM [9]	31.81	16.22	06.00	76.56	42.15	28.21
RU-LSTM [16]	33.04	22.78	14.39	79.55	50.95	33.73
Temp. Agg. [52]	**37.87**	24.10	**16.64**	**79.74**	53.98	**36.06**
MM-Sep (HO+ST+M+BOO)	28.59	**27.18**	10.85	78.64	**57.66**	30.83
(c) Test Set - Unseen Kitchens (S2)						
Pred+Trans [41]	28.40	12.40	07.20	69.80	32.20	19.30
TSN+SVM [9]	25.30	10.41	02.29	68.32	27.38	09.35
RU-LSTM [16]	27.01	15.19	08.16	69.55	34.38	21.10
Temp. Agg. [52]	**29.50**	16.52	**10.04**	70.13	37.83	**23.42**
MM-Sep (HO+ST+M+BOO)	26.80	**18.40**	06.76	**70.40**	**44.18**	20.04

H. Anticipating actions in long-term future

Long-term anticipation is especially useful in planning for actions that can help prevent mishaps in manufacturing scenarios or assisted living scenarios. Hence, we consider anticipation times of 1, 2, and 3 minutes while keeping the observation fixed to 1 second. We choose the 50Salads dataset with relatively long videos of around 6.4 minutes on average and 20 action instances per video for this task. The results are presented in Table XIII, and we compare with other temporal networks like GRU+CRF, MS-TCN, and TRN. All the models were trained for the respective anticipation periods and then evaluated. Human-object features perform consistently well when paired with different temporal networks for long-term anticipation. The performance of the temporal networks on long-term anticipation follows the same trend as immediate anticipation (Table IV), with transformers performing better than other networks. Interestingly, both human-object based and multi-modal transformer predict actions far into the future with similar accuracy. So, we can conclude that the representation of human-object interactions can provide enough information for effective long-range anticipation.

TABLE XIII
LONG-TERM ACTION ANTICIPATION ACCURACY ON 50 SALADS DATASET

Method	Anticipation Period		
	1 min	2 min	3 min
HO(CC) + GRU+CRF	19.1	16.5	15.6
HO(CC) + MS-TCN	29.3	27.1	23.7
HO(CC) + TempRec	32.6	29.1	27.5
HO(CC) + TempRelNet	32.8	29.5	27.9
HO(CC) + Transformer	34.1	32.1	30.1
Multi-modal Transformer (MM-Sep)	**34.3**	**32.3**	**30.3**

V. DISCUSSION AND CONCLUSION

In this work, we addressed the problem of anticipation of actions using pairwise human-object interactions by considering only their visual features. We proposed cross-correlation as a way to capture higher-order statistics across human-object pairs in a frame. We showed that cross-correlation achieves better action anticipation compared to both attention and covariance pooling on scripted datasets. Further, our experiments showed that for the proposed approach, relative proximity between humans and objects is not as crucial for anticipation when using cross-correlation. In certain frames, objects are not detected due to either occlusion (Breakfast) or camera motion in egocentric videos (EPIC-KITCHENS55). As action instances in both Breakfast and EPIC-KITCHENS55 are generally long in duration, we can recover at least 1 second of observed frames (15 for Breakfast or 30 for EPIC-KITCHENS55) to obtain relevant information. Also,

sudden appearance and disappearance of objects in EPIC-KTICHENS55 can be handled using cross-correlation as we can obtain a fixed-dimensional representation from variable number of human-object relations per frame.

Different temporal networks were evaluated for action anticipation with pairwise human-object interaction representation, and transformer was found to be the most effective. Our ablation studies showed that use of transformer models at both encoder and the decoder improves the action anticipation performance on both 50 Salads and Breakfast datasets. We also proposed two approaches to build multi-modal Transformer that can leverage the evidence across different frame representations. Multi-modal transformer is trained using human-object, spatio-temporal, and motion features and it showed improved performance than existing methods on both 50Salads and Breakfast datasets. We were able to obtain good long-term anticipation performance by using pairwise human-object representation as well, including when we predict 3 minutes into the future. Finally, we can conclude that using only visual features with cross-correlation is sufficient in representing human-object interactions and anticipating actions in the immediate and long-term future.

One limitation of our current model is that it seems our model performs exceptionally well on scripted datasets, while the performance on unscripted datasets is not that convincing except for out-of-domain evaluation such as test set 2 (S2) of EPIC-KTICHENS55 dataset and good noun prediction performance. In the future, other modalities like depth information and language models can be explored in conjunction with human-object pairs for better action anticipation.

ACKNOWLEDGMENT

This research/project is supported in part by the National Research Foundation, Singapore under its AI Singapore Program (AISG Award No: AISG2-RP-2020-016) and the National Research Foundation Singapore under its AI Singapore Program (Award Number: AISG-RP-2019-010).

REFERENCES

[1] Y. Abu Farha, A. Richard, and J. Gall. When will you do what?-anticipating temporal occurrences of activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5343–5352, 2018.

[2] D. Acharya, Z. Huang, D. Pani Paudel, and L. Van Gool. Covariance pooling for facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 367–374, 2018.

[3] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 56(2):411–421, 2006.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of 3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[5] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori. Object level visual reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 105–121, 2018.

[6] L. Berrada, A. Zisserman, and M. P. Kumar. Smooth loss functions for deep top-k classification. In *Proceedings of the International Conference on Learning Representations*, 2018.

[7] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun. Anticipating accidents in dashcam videos. In *Asian Conference on Computer Vision*, pages 136–153. Springer, 2016.

[8] A. Cherian and S. Gould. Second-order temporal pooling for action recognition. *International Journal of Computer Vision*, 127(4):340–362, 2019.

[9] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[10] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020.

[11] M. Engin, L. Wang, L. Zhou, and X. Liu. Deepkspd: Learning kernel-matrix-based spd representation for fine-grained image recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 612–627, 2018.

[12] Y. A. Farha and J. Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019.

[13] B. Fernando and S. Herath. Anticipating human actions by correlating past with the future with jaccard similarity measures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[14] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[15] A. Furnari, S. Battiato, and G. Maria Farinella. Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, 2018.

[16] A. Furnari and G. M. Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6252–6261, 2019.

[17] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes. Forecasting future action sequences with neural memory networks. In *Proceedings of the 30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 298. BMVA Press, 2019.

[18] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes. Predicting the future: A jointly learnt model for action anticipation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5562–5571, 2019.

[19] K. Gavrilyuk, R. Sanford, M. Javan, and C. G. Snoek. Actor-transformers for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 839–848, 2020.

[20] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. Video action transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019.

[21] G. Gkioxari, R. Girshick, P. Dollár, and K. He. Detecting and recognizing human-object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8359–8367, 2018.

[22] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth International Conference on artificial intelligence and statistics*, pages 315–323, 2011.

[23] S. Halder, K. H. Lim, J. Chan, and X. Zhang. Transformer-based multi-task learning for queuing time aware next poi recommendation. *Journal: Advances in Knowledge Discovery and Data Mining Lecture Notes in Computer Science*, pages 510–523, 2021.

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[25] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[26] C. Ionescu, O. Vantzos, and C. Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2965–2973, 2015.

[27] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.

[28] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid. Joint learning of object and action detectors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4163–4172, 2017.

[29] Q. Ke, M. Fritz, and B. Schiele. Time-conditioned action anticipation in one shot. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9925–9934, 2019.

[30] D.-J. Kim, X. Sun, J. Choi, S. Lin, and I. S. Kweon. Detecting human-object interactions with action co-occurrence priors. In *European Conference on Computer Vision*, pages 718–736. Springer, 2020.

[31] H. Koppula and A. Saxena. Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. In *Proceedings of the International Conference on machine learning*, pages 792–800, 2013.

[32] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, 2015.

[33] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 780–787, 2014.

[34] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks for action segmentation and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[35] J. Libovickỳ, J. Helcl, and D. Mareček. Input combination strategies for multi-source transformer decoder. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 253–260, 2018.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[37] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457, 2015.

[38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[39] A. Mallya and S. Lazebnik. Learning models for actions and person-object interactions with transfer to question answering. In *European Conference on Computer Vision*, pages 414–428. Springer, 2016.

[40] P. Mettes and C. G. Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4443–4452, 2017.

[41] A. Miech, I. Laptev, J. Sivic, H. Wang, L. Torresani, and D. Tran. Leveraging the present to anticipate the future in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[42] T. Nagarajan, Y. Li, C. Feichtenhofer, and K. Grauman. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 163–172, 2020.

[43] Y. B. Ng and B. Fernando. Forecasting future action sequences with attention: a new approach to weakly supervised action forecasting. *IEEE Transactions on Image Processing*, 29:8880–8891, 2020.

[44] M. Oliu, J. Selva, and S. Escalera. Folded recurrent neural networks for future video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 716–731, 2018.

[45] X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.

[46] S. Qi, W. Wang, B. Jia, J. Shen, and S.-C. Zhu. Learning human-object interactions by graph parsing neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 401–417, 2018.

[47] V. Ramanathan, C. Li, J. Deng, W. Han, Z. Li, K. Gu, Y. Song, S. Bengio, C. Rosenberg, and L. Fei-Fei. Learning semantic relationships for better action retrieval in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1100–1109, 2015.

[48] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. Pedestrian action anticipation using contextual feature fusion in stacked rnns. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 171. BMVA Press, 2019.

[49] C. Rodriguez, B. Fernando, and H. Li. Action anticipation by predicting future dynamic images. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[50] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Proceedings of the International Conference on Computer Vision*, pages 1036–1043. IEEE, 2011.

[51] M. Sadegh Aliakbarian, F. Sadat Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson. Encouraging lstms to anticipate actions very early. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 280–289, 2017.

[52] F. Sener, D. Singhania, and A. Yao. Temporal aggregate representations for long-range video understanding. In *Proceedings of European Conference on Computer Vision*, pages 154–171. Springer, 2020.

[53] Y. Shi, B. Fernando, and R. Hartley. Action anticipation with rbf kernelized feature mapping rnn. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 301–317, 2018.

[54] B. Soran, A. Farhadi, and L. Shapiro. Generating notifications for missing actions: Don't forget to turn the lights off! In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4669–4677, 2015.

[55] S. Stein and S. J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM International joint Conference on Pervasive and ubiquitous computing*, pages 729–738, 2013.

[56] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid. Actor-centric relation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–334, 2018.

[57] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[58] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh. Anticipating traffic accidents with adaptive loss and large-scale incident db. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3521–3529, 2018.

[59] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *IEEE transactions on pattern analysis and machine intelligence*, 30(10):1713–1727, 2008.

[60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[62] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–106, 2016.

[63] B. Wan, D. Zhou, Y. Liu, R. Li, and X. He. Pose-aware multi-level feature network for human object interaction detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9469–9478, 2019.

[64] B. Wang, L. Huang, and M. Hoai. Active vision for early recognition of human actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1081–1091, 2020.

[65] Q. Wang, J. Xie, W. Zuo, L. Zhang, and P. Li. Deep cnns meet global covariance pooling: Better representation and generalization. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[66] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[67] X. Wei, P. Lucey, S. Vidas, S. Morgan, and S. Sridharan. Forecasting events using an augmented hidden conditional random field. In *Proceedings of the Asian Conference on Computer Vision*, pages 569–582. Springer, 2014.

[68] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[69] M. Xu, M. Gao, Y.-T. Chen, L. S. Davis, and D. J. Crandall. Temporal recurrent networks for online action detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5532–5541, 2019.

[70] L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual madlibs: Fill in the blank description generation and question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2461–2469, 2015.

[71] Y. Zhang, P. Tokmakov, M. Hebert, and C. Schmid. A structured model for action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9975–9984, 2019.

[72] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[73] B. Zhou, A. Andonian, A. Oliva, and A. Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018.