Pulling Actions out of Context: Explicit Separation for Effective Combination

Yang Wang and Minh Hoai
Stony Brook University, Stony Brook, NY 11794, USA
{wang33, minhhoai}@cs.stonybrook.edu

Abstract

The ability to recognize human actions in video has many potential applications. Human action recognition, however, is tremendously challenging for computers due to the complexity of video data and the subtlety of human actions. Most current recognition systems flounder on the inability to separate human actions from co-occurring factors that usually dominate subtle human actions.

In this paper, we propose a novel approach for training a human action recognizer, one that can: (1) explicitly factorize human actions from the co-occurring factors; (2) deliberately build a model for human actions and a separate model for all correlated contextual elements; and (3) effectively combine the models for human action recognition. Our approach exploits the benefits of conjugate samples of human actions, which are video clips that are contextually similar to human action samples, but do not contain the action. Experiments on ActionThread, PASCAL VOC, UCF101, and Hollywood2 datasets demonstrate the ability to separate action from context of the proposed approach.

1. Introduction

A human action does not occur in an isolated vacuum tube, and it is not the only thing recorded in a video. A video clip of a human action is the melting pot of the action and various other ingredients including the background scene, the object tools, the camera motion, the lighting condition, the other body movements and actions. Many of these components are totally unrelated to the action, while some are contextual elements that frequently co-occur with the category of action in consideration. For example, a cooking action usually occurs in a kitchen so the kitchen scene is a relevant *context*, but the rare incident of a bird landing on a windowsill that also appears in the video would be *noise*. Neither noise nor context corresponds to the actual content of a human action, but they have different impacts. Noise always hurts the performance of a recognition system, while context may provide useful cues for recognition. For example, recognizing the kitchen scene of a cooking action will prevent it from being misclassified as an outdoor activity such as surfing or golfing. Thus context is informative and can be used to aid recognition. But context can also confound recognition algorithms. The existence of context dilutes the actual content of an action and makes recognition difficult, especially for fine-grain categorization between action categories that share similar or the same context. For example, a hair-combing action in a bathroom might be mistakenly recognized as toothbrushing due to background similarity.

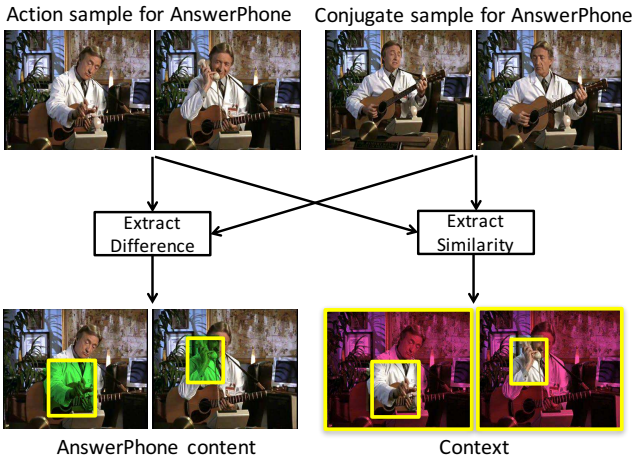

Figure 1. How do we obtain an action classifier that *focuses* on the *action* components and not the context? Unfortunately, such a classifier cannot be obtained by the normal supervised learning approach, because training examples of human actions do not generally come with detailed human annotation that delineates the subtle human actions (from the co-occurring context). To address this problem, we propose to collect video sequences that are contextually similar to the action samples, which will be referred to as conjugate samples. We propose to use conjugate samples to train a classifier that deliberately separate action from context. Best viewed on a digital device.

So, how can we recognize a subtle human action in the presence of noise and context? A typical approach is to ignore the distinction of these factors and use supervised machine learning to train a classifier to separate positive training examples (i.e., video clips containing the action in

consideration) from negative ones (video clips that do not contain the action). There is an optimistic hope that the classifier can learn a model of the human action based on the commonalities in positive training examples that never or seldom exist in negative training examples. While this approach can suppress noises which are rare incidents in positive training examples, it cannot effectively separate the actual human action from the frequently co-occurring contextual elements. For example, if many training examples of the toothbrushing action have a bathroom background, the trained classifier might look for bathroom cues instead of the actual toothbrushing motion. Similarly, a *kissing* classifier that is trained using video clips from Hollywood movies may attend to elements that have nothing to do with the kiss itself, such as the camera angle and scene illumination. Unfortunately, failure to factorize human actions from context has severe consequences. First, due to the dominance of context in a video, it would be difficult for fine-grain classification between action categories that share similar context. Second, the learned classifier will fail to generalize to real-world applications where the context is different. Furthermore, for the purpose of understanding human actions and interpreting the classifier's decision, the inability to separate the action from context is deeply unsatisfactory.

Then how do we separate human actions from context? One possible thought is to train two separate classifiers, one to recognize the action content and the other for the contextual elements. This approach, however, requires training data with detailed annotation, one in which the actual content of human actions, the contextual elements, and the irrelevant noisy incidents are all annotated. Collecting manual annotation at this level of details is notoriously difficult, if not impossible. Furthermore, this approach is not scalable to a large system where we need to recognize thousands of human actions.

In this paper, we propose a novel approach to human action recognition, one that can explicitly factorize human actions from context. Our key idea is to exploit the benefits of the information from *conjugate samples* of human actions. Here we define a conjugate sample as a video clip that is *contextually similar* to an action sample, but does not contain the action. For instance, a conjugate sample of an "answer phone" can be the video sequence showing the person approaching the phone or doing another action prior to answering the phone (e.g., playing guitar as in Figure 1). The answer phone clip and the video sequence preceding it have many similar or even the same contextual elements, including the people, the background scene, the camera angle, and the lighting condition. The only thing that sets the two clips apart is the actual human action itself. A conjugate sample provides contrasting information to the action sample; it can be used to suppress contextual irrelevance and magnify the action signal, as illustrated in Fig. 1.

The context that we refer to in this paper is more than the background or the scene category. It is defined as any visual element that is often observed with the action but does not correspond to the actual motion of the action. Such context could refer to the camera movement, the camera angle, the illumination condition, the pose of the actor, the configuration of people, the social norm in a group interaction (e.g., a handshake often occurs in a meet-and-greet situation), or the sequential order of actions (e.g., a handshake normally follows an arm extension).

2. Related Work

The benefits of context for human action recognition are well recognized and have been confirmed by many studies. Various contextual elements have been considered, including scene categories (e.g., [18, 20, 21, 30, 44]), objects (e.g., [3, 4, 7–10, 10, 15, 15, 16, 20, 22, 39, 42]), pose and people configuration (e.g., [4, 13, 17, 31, 34, 40, 41, 43]), group context and social roles (e.g., [2, 19, 24]), temporal context (e.g., [1, 28, 33]), and context from other action categories (e.g., [12, 14]). However, the usage of context in these works is fundamentally different from what is being proposed here. Many existing methods, e.g., [10, 30], rely on local features that are customized to capture the contextual elements of interest, such as scene and object descriptors. These local descriptors are computed densely for the entire video clip, making no attempt to distinguish between action and context. There are methods that learn separate models for action and context, e.g., [1, 20, 21], but the action model is not explicit to the action content and the context model requires additional annotated training data. The proposed approach will be the first to explicitly factorize action from context, deliberately model each of them, and effectively combine them for recognition. All of these steps are jointly optimized in an integrated framework.

3. Action-Context Separation & Combination

Our goal is to explicitly separate subtle human actions from the dominance of co-occurring context. The key idea is to collect conjugate samples of human actions and develop a framework for this novel type of training data.

3.1. Collecting conjugate samples of human actions

We propose to collect conjugate samples for each human action sample based on temporal proximity, that is to use video clips before and after the action sample. The video sequences right before and after an action sample are excellent conjugate samples. Many contextual elements of these video sequences and the action sample are very similar or even the same, from the background scenes and the actors to the camera angles and the lighting conditions. The only thing that sets them apart is the actual human action.

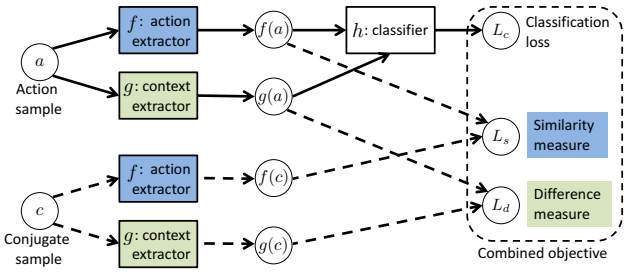

Figure 2. **Using conjugate samples for training a human action classifier.** This is a deep multi-stage architecture where feature extraction and classifier learning are two stages of a joint learning framework. Functions f and g are for extracting the action and context feature vectors respectively. The extracted feature vectors are subsequently fed into the classifier h. The objective is to minimize the classification loss and the similarity between two action feature vectors $f(a)$ and $f(c)$, while maximizing the similarity between two context feature vectors $g(a)$ and $g(c)$. Note that the dotted lines are only effective in the learning phase and removed at test time.

3.2. Proposed framework – general approach

Conjugate samples provide contrasting information to the action samples, but how should we maximize their benefits? A naive approach is to treat them as negative training examples. This approach, however, is unlikely to yield good performance. A positive action sample and its corresponding conjugate samples are contextually similar, so treating conjugate samples as negative training data would force contextual cues as negative evidence. However, contextual cues are crucial for categorization, especially for separating dissimilar classes such as distinguishing between hugging and surfing. The second naive approach is the exact opposite of the first one, treating all conjugate samples as positive training data. However, conjugate samples do not contain the action of interest, so this approach will learn contextual cues instead of the actual human action. For example, toothbrushing often occurs in a bathroom, and the system being trained might look for bathroom cues instead of the actual toothbrushing motion. Another approach is to associate each conjugate sample with a latent variable and then infer the label in the training process. This approach will lead to a mixture of positive and negative class labels, failing to separate actions from context.

We propose here a framework for integrating conjugate samples in the training process without the need to assign or infer the class label to each conjugate sample. It is a multi-task learning framework where the conjugate samples are only used for separating the action content from the dominating context. Figure 2 illustrates the architecture of the learning framework. This architecture has two major stages for feature extraction and classifier learning, which are jointly optimized. The input to the network is a pair of an action sample a and a conjugate sample c. Functions f and g are for extracting the action and context feature vectors respectively. The extracted feature vectors are subsequently fed into a classifier h. Generally, f, g, and h can be any function (e.g., a multi-layer neural network), but the forms must be fixed during training while the parameters of the functions are to be learned. These parameters can be learned by minimizing a combined objective function, which consists of: i) the classification loss; ii) the similarity measure between two action feature vectors; and iii) the dissimilarity measure between two context feature vectors. Our assumption is that the conjugate sample is contextually similar to the action sample, therefore the context extractor function g should yield very similar feature vectors. On the other hand, what distinguish between the action sample and the conjugate sample is the actual content of the action. Thus the action extractor function f should yield very different feature vectors. Notably, the classification loss only depends on the action sample. Because both action and context are important for action recognition, the classifier depends on both the action feature vector and the context feature vector. The key novelty here is the explicit separation between action and context, which enables the classifier to selectively use the context only when necessary. Another benefit of the separation between action and context is visual interpretability. By tracing the classifier decision, we can understand the important factors that leads to the decision of the classifier, whether they are attributed to the action or the context.

It should be noted that the conjugate samples are only needed in the training phase. Once trained, the action classifier can be used to predict the label of any test video clip without any conjugate sample. Furthermore, conjugate samples are not required for all action samples. If an action sample does not have a corresponding conjugate sample, the loss term for the action sample can be simplified to the classification loss. It is also possible for an action sample to have multiple conjugate samples. In this case, the similarity and difference metrics L_s and L_d can be chosen to measure the similarity and difference between a vector and a set of vectors instead.

3.3. Proposed framework – a specific instance

The general learning framework described above is flexible. It can be used with different models of action and context extractor functions. The framework can also be used with different forms of loss measures. In this section, we describe a simple instance of the general framework, which will be thoroughly evaluated in the experiment section.

Architecture. The particular framework is based on C3D, a 3D convolutional neural network for human action recognition [29]. It assumes that both the action samples and conjugate samples are represented by a 16-frame volume. The

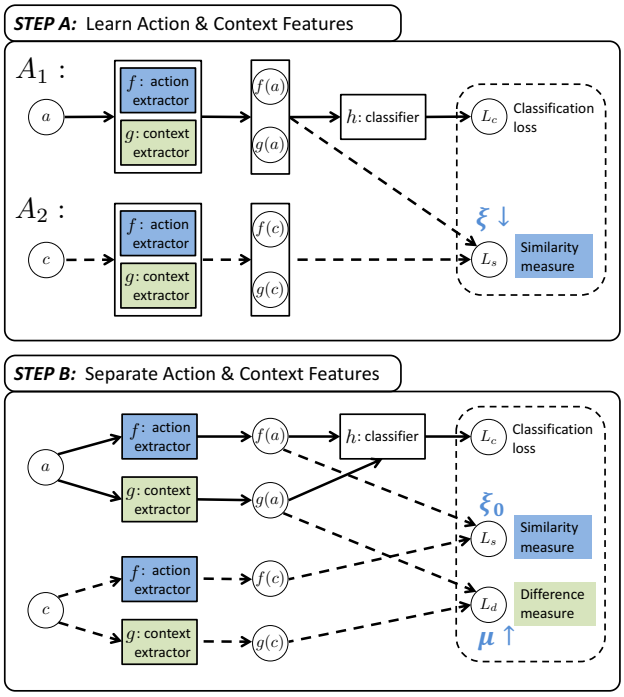

Figure 3. **Two-stage optimization of the proposed framework.** Step A: Learning action and context features for recognition task; Step B: Learning to separate action and context features using conjugate samples. The optimization procedure is further detailed in Table 1. Note that the dotted lines are only effective in the learning phase and removed at test time.

C3D network uses several types of layers namely convolution, max pooling, fully connected, rectified linear unit, drop-out, and soft-max. Let $C(n)$ denote a convolution layer with n kernels (kernel size: $3\times3\times3\times depth$) followed by rectified linear units, M a max pooling layer, $FC(n)$ a fully connected layer with n filters followed by rectified linear units, $D(r)$ a dropout layer with dropout ratio r, and SF a soft-max layer. Suppose action and conjugate samples are both 3D volumes with RGB channels ($112 \times 112 \times 3 \times 16$) and there are k action classes, the architecture of our network is as follows. Both the action and context extractors f and g are: $C(64) \rightarrow M \rightarrow C(128) \rightarrow M \rightarrow C(256) \rightarrow C(256) \rightarrow M \rightarrow C(512) \rightarrow C(512) \rightarrow M \rightarrow C(512) \rightarrow C(512) \rightarrow M \rightarrow FC(2048)$. The classifier h is $D(0.5) \rightarrow FC(4096) \rightarrow D(0.5) \rightarrow FC(k) \rightarrow SF$. We initialize the components f, g, and h using the weights from a published C3D model [29] pre-trained on Sports1M dataset. For networks that take still images as input, we use VGG16 model [26] instead of the C3D model.

Loss Function. For a training pair of action sample and conjugate sample, the loss is $L = L_c + L_s + L_d$, which is the sum of the classification loss, the similarity loss between action vectors, and the difference loss between context vectors. For classification loss L_c, we choose the soft-max log-loss, which is a commonly used criterion in multi-class classification. To measure the similarity or difference between two vectors, we first perform batch-normalization to each feature channel so that all feature channels have zero mean and unit variance, thus contribute equally to the distance measure. We will refer to the normalized action and context feature vectors as $\bar{f}(\cdot)$ and $\bar{g}(\cdot)$ respectively. For the similarity and difference losses L_s and L_d, we use cosine distance, which is length-invariant and robust. Specifically:

$$L_s = \lambda \cdot \max\left\{0,\ cos\langle \bar{f}(a), \bar{f}(c)\rangle - \xi\right\} \quad (1)$$

$$L_d = \mu \cdot \left(1 - cos\langle \bar{g}(a), \bar{g}(c)\rangle\right). \quad (2)$$

The loss L_s measures the similarity between two action vectors extracted from a pair of action and conjugate samples. The parameter λ is typically set to either 0 or 1, depending on whether L_s should be included in the total loss. The parameter ξ is the main tunable parameter for the similarity loss; it is the threshold for penalization. For example, if $\xi = 0.5$, the similarity loss would drop to 0 if the angle between $\bar{f}(a)$ and $\bar{f}(c)$ is more than 60°. When $\xi = -0.5$, that would require $\bar{f}(a)$ and $\bar{f}(c)$ to be 120° apart. Thus a smaller ξ would impose more pressure to push the two action components apart. The loss L_d is to minimize the difference between the two context vectors, and μ is a tunable parameter.

Optimization. In general, our final network uses $\lambda = 1$ and a big value for μ and a small value for ξ, emphasizing the importance for separating action and context. However, directly optimizing a network with strong regularization parameters (big μ, small ξ) may lead to a bad local minimum. We therefore propose a two-stage training procedure depicted in Figure 3 and detailed in Table 1. At Step A_1, we first train the baseline network in the absence of conjugate samples ($\lambda = 0$). This step ensures the network learn useful action/context cues for the classification task. However the action and context features are mixed, because the network has no intention or means to identify and factorize them. At Step A_2, we aim to improve the network's performance by leveraging the contrasting information from the conjugate samples. We start the procedure with weak regularization parameters (e.g., $\lambda = 1,\ \xi = 0.3$), and gradually decrease the value of ξ to segregate the feature vectors for a pair of action and conjugate samples. This process usually leads to better performance in our experiments. Finally, at Step B, we divide the C3D network into two channels: the action channel f and the context channel g, and fine-tune it with a fixed ξ_0 and a gradually increased μ.

From Step A_2 to B, we use a simple procedure to divide the neurons of the FC6 layer into disjoint subsets of action and context neurons, while freezing all the shared convolutional layers preceding the FC6 layer. First, for each neuron n at FC6 layer, we calculate the accumulated activation gap

Step	(λ	ξ)	μ	Purpose
A_1	(0	)		train baseline network
A_2	(1	$0.3\downarrow$)		improve performance
B	(1	ξ_0)	$0\uparrow$	separate action & context

Table 1. **Optimization procedure**. Two steps to factorize the action and context components. Step A_1 & A_2 train a network to focus on recognition performance and initiate the feature separation. Step B purifies both the action extractor and the context extractor.

$\delta(n) = \sum_i n(a_i) - n(c_i)$, where $n(a_i)$ and $n(c_i)$ are the activation values of neuron n, given a pair of action sample a_i and conjugate sample c_i from the i^{th} training video. The accumulated activation gap is a good indicator to initialize the separation between an action and context neurons. Second, we rearrange and divide the neurons based on their accumulated activation gaps as follows. The FC6 layer has a total of 4096 neurons, each corresponds to a single row in the weight parameter $\mathbb{W} \in R^{4096\times N}$ and the bias parameter $\mathbb{B} \in R^{4096\times 1}$. The rows of $\mathbb{W}$ and $\mathbb{B}$ are rearranged into $\mathbb{W}'$ and $\mathbb{B}'$ such that a neuron with a larger $\delta(n)$ (more action-related) would have a smaller row-index. The rows of $\mathbb{W}'$ and $\mathbb{B}'$ are then split into two equal groups. The first corresponds to action features, and the second to context features. During these steps, the columns of the FC7 weight parameter are also reordered and divided accordingly to ensure correct correspondence with neurons at FC6 layer.

Implementation. Our network is implemented in Torch deep-learning library. The network is trained using back-propagation with mini-batch stochastic gradient descent. We use a fixed batch size of 64, a learning rate of 0.001, a momentum of 0.9, and a weight decay of 0.0005.

Action and context activation. Once a network has been trained, we can consider the activation of the neurons right before the soft-max layer. This is a score vector of which the size is the number of classes. Each element of the score vector corresponds to a class, and it is the sum of the *action score*, the *context score*, and a class-specific offset value. The action score corresponds to the activation strength of the action extractor, while the context score is the activation strength of the context extractor. We will analyze the action and context activation strengths in our experiments below.

4. Experiments

4.1. Separating action and context in video

The experiments in this section is performed on the ActionThread dataset [11]. This dataset contains video clips that include human actions as well as the sequences before and after the action. This allows us to collect conjugate samples of human actions, which is why we use ActionThread instead of other more popular datasets such as Hollywood2 [21] and TVHID [23]. This dataset has 3035 video clips of 13 different actions, which are split into disjoint train and test subsets [11]. We consider the pre- and post-action sequences as the source of conjugate samples, and ensure the action and conjugate samples are extracted from the same thread and by the same cropping window.

C3D features. We use C3D features [29] as video representations. Compared to other state-of-the-art methods such as Dense Trajectory Descriptors (DTD) [32] or Two-stream CNN [25], C3D achieves a good balance between efficiency and simplicity. DTD produces very high-dimensional (∼100k-dim) Fisher vectors, and Two-stream CNN requires heavy computation for extracting optical flow images, whereas C3D network only requires RGB input and gives compact (4096-dim) representations. After training a C3D model, to extract the features, a video is split into 16-frame-long clips with a 8-frame overlap between two consecutive clips. We then feed these clips into the C3D network to extract FC6 activations, which are temporally aggregated to form a single video descriptor and subsequently L2 normalized. We use eigen evolution pooling for temporal aggregation of C3D features because it consistently outperforms average pooling in our experiments. We refer the reader to [29, 37, 38] for more details.

Action recognition. Table 2 compares the performance of several methods for recognizing human actions in the ActionThread dataset. Compared with the baseline method that does not use conjugate samples of human actions, our method achieves significantly better performance, and the improvement for some classes such as *AnswerPhone*, *Fight*, *ShakeHand*, and *Hug* is very significant, as high as 15%. Meanwhile, the two alternative approaches of using conjugate samples as either negative or positive training examples lead to lower mean average precision.

Table 3 shows the benefits and also complimentary benefits of the proposed method with the state-of-the-art methods on ActionThread using C3D and DTD feature descriptors. The proposed method (Factor-C3D) outperforms the baseline method C3D by a margin of 6%; this is the direct comparison for the benefits of the proposed approach because both methods use the same feature descriptors and the only difference is whether conjugate samples are used. The proposed method also outperforms DTD, and the complementary benefit is 22% relative AP improvement.

Role of context features. The role of context features for action recognition can be investigated with our network, owing to its ability to explicitly factorize features into the action component $f(a)$ and the context component $g(a)$. We study the effect of the context component by removing it from the combined feature vector and measuring the change in action recognition performance, as depicted in Figure 4.

	How conjugate samples are used			
	NotUsed [29]	AsNegative	AsPositive	Proposed
AnswerPhone	27.3	27.3	28.7	**43.0**
DriveCar	51.2	51.6	48.9	**53.1**
Eat	36.9	37.7	35.7	**42.1**
Fight	45.7	48.6	41.2	**61.1**
GetOutCar	29.7	**31.4**	30.1	27.2
ShakeHand	26.9	26.0	26.6	**35.2**
Hug	43.9	44.1	45.5	**54.7**
Kiss	67.4	66.4	67.0	**72.6**
Run	82.0	80.6	79.9	**85.6**
SitDown	36.3	35.7	36.0	**45.2**
SitUp	**17.1**	13.4	15.2	15.7
StandUp	**31.9**	31.0	31.4	28.5
HighFive	49.3	40.6	48.8	**58.5**
Mean	42.0	41.1	41.2	**47.9**

Table 2. **Action recognition results on the ActionThread dataset.** The table shows average precision values; a higher number indicates a better performance. All four settings use the same C3D architecture [29]. NotUsed is the baseline method that does not use conjugate samples. AsNegative and AsPositive are the methods that use conjugate samples as negative and positive training examples respectively. Our method achieves significantly better performance than the other methods.

Method	No Pruning	Pruning
Pretrained C3D [29]	35.4	-
Finetuned C3D [29]	42.0	-
Factor-C3D [Proposed]	**47.9**	-
DTD [32]	45.3	52.1
Non-Action [36]	48.0	55.0
DTD + C3D	52.0	56.3
DTD + Factor-C3D [Proposed]	**55.3**	**58.5**

Table 3. **Benefits and complementary benefits of the proposed method with other state-of-the-art methods.** 'Pruning' means the non-action classifier is used. The fairest comparison is between Factor-C3D and Finetuned C3D because they both use the same feature descriptors. Factor-C3D provides a large complementary benefit to DTD.

For instance, for classifying between two contextually similar actions *Kiss* and *Hug*, removing the confounding context features (i.e., using $f(a)$ instead of $f(a)+g(a)$) leads to better classification result. However, for two actions with very different context, e.g., *Kiss* and *Eat*, the classifier performs worse when the context component is removed, because the context is useful for separating contextually dissimilar actions. These experiments show that the context should not be blindly used or removed. Instead, we can determine the importance of context with a weighting scheme

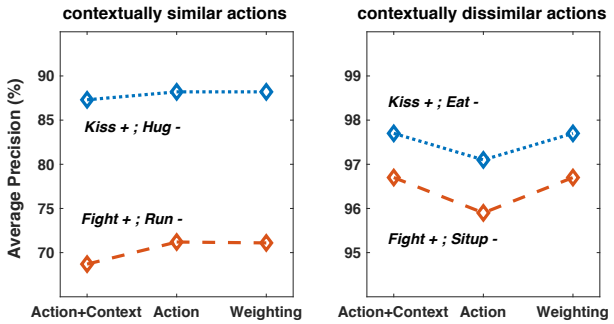

Figure 4. **Performance change for binary classification after removing context features.** +/- stands for positive/negative samples. Left: for distinguishing between actions with similar context, removing the confounding context component improves the classification results. Right: for distinguishing contextually dissimilar actions, removing the context component decreases the performance. These results show the importance of context, but context should not be blindly used or removed. It is beneficial to use context selectively, which requires explicit action-context separation as proposed here. Given the explicit separation, we can tune the amount of context to use in a weighted combination, as successfully done here.

$f(a)+\gamma g(a)$, where γ is a tunable parameter. This is effective, as shown in Figure 4, where γ is tuned using validation data. For this experiment, we divide the videos in the test set into disjoint test/validation subsets using 80/20 split.

Visualizing action and context components. We also visualize the video clips that excite the action and context extractors the most. For each video in the test set, We sample 20 clips from the action sequence and another 20 from the pre- or post-action sequences. We feed each clip into our final network and measure the action and context activation strengths separately. For each action class, we identify the video clips that have the highest action and context activation strengths. Figure 5 depicts some of these video clips for actions *AnswerPhone, Hug, ShakeHand* and *SitUp*. As can be seen, the most helpful contextual cues for recognizing *Hug* and *ShakeHand* appear to be a group of humans, whereas bed scene is contextually associated to action *SitUp*. This is the visual evidence that our network has indeed learned to factorize action and context components.

Pairwise context difference. Since we have the context extractors for every class, we can analyze the pairwise context similarity between action classes. Consider the context extractor of Class R and all action samples of Class X, we calculate the mean activation of the context extractor and let $c(X,R)$ denote this quantity. Using $c(R,R)$ as a reference, we consider the context difference between Class R and Class X as: $d(X,R) = c(R,R) - c(X,R)$. The context difference indicates the amount of context similarity between two classes. Figure 6 shows the context differ-

Figure 5. **Representative patterns with highest action and context activation values.**

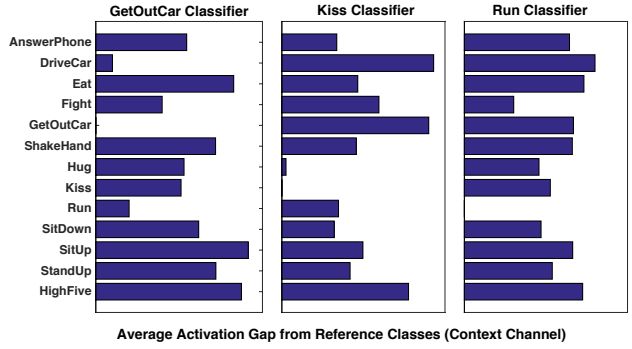

Figure 6. **Pairwise context difference** between GetOutCar (left), Kiss (middle), and Run (right) with other action classes. Smaller context difference indicates higher context similarity. In terms of contextual similarity, DriveCar is close to GetOutCar, Hug is close to Kiss, while Fight is close to Run.

ence for three action classes: GetOutCar, Kiss, and Run. As can be seen, GetOutCar is contextually similar to DriveCar, Kiss is similar to Hug, and Fight is similar to Run.

4.2. Imperfect conjugate samples

So far, we have assumed that we can retrieve perfect conjugate samples that do not contain the target human actions in consideration. This assumption is true in general, unless we are forced to exclusively work with a dataset that have no corresponding pre- or post-action sequences such as UCF101 [27] and Hollywood2 [21]. We question whether it is necessary to have a black-and-white separation between conjugate and action samples with respect to the action content. We want to study the benefits and drawbacks of the proposed framework when the action samples may not contain the entire human action and the conjugate samples cannot be guaranteed to exclude all of the action. In this section, we describe the experiments to study this scenario, using UCF101 [27] and Hollywood2 [21] datasets.

Given a training video in either UCF101 or Hollywood2 dataset, we extract a sequence of 16 video frames and use it as the action sample. From the same video, we extract another sequence of 16 frames before or after the action sample to create the corresponding conjugate sample. These two video samples have the same context, and what distinguish them is the difference between the two dynamics stages of a human action.

Method	UCF101	Hollywood2
C3D [29]	82.3	49.8
Factor-C3D [Proposed]	**84.5**	**54.7**
DTD [32]	85.9	67.5
DTD + C3D	90.8	69.5
DTD + Factor-C3D [Proposed]	**91.3**	**71.3**
EigenTSN [37]	95.3	75.5
EigenTSN + C3D	**95.8**	76.1
EigenTSN + Factor-C3D [Proposed]	**95.8**	**76.7**

Table 4. **Action recognition results on UCF101 and Hollywood2.** The comparison between Factor-C3D and C3D is the direct measurement for the advantage of the proposed framework with conjugate samples. State-of-the-art performance can be achieved when combining the proposed method with others. We report accuracy for UCF101 and mean AP for Hollywood2.

Using the generated conjugate samples, the proposed framework achieves a mean average precision of 54.7% on the Hollywood2 dataset, outperforming the baseline C3D network (49.8%) where conjugate samples are not used. On the three splits of the UCF101 dataset, the proposed framework achieves an average accuracy of 84.5%, outperforming the baseline C3D network (82.3%) that does not use the conjugate samples. On both Hollywood2 and UCF101, the performance gains for using conjugate samples are significant, even though the approach of generating conjugate samples is not ideal. The performance gains can be attributed to the ability of the framework to force the action extractor to focus on the dynamics of the action, rather than the scene context. On the other hand, the performance gains on Hollywood2 and UCF101 are not as high as the performance gain obtained on the ActionThread dataset, where we could extract proper conjugate samples.

The results reported in previous paragraph should not be compared directly to the highest reported numbers in previous publications, which have been obtained by combining multiple features and methods [6, 35]. The proposed method provides complementary benefits to other methods, and the state-of-the-art results can be achieved by combining them, as shown in Table 4.

4.3. Separating action and context in still images

The proposed framework for separating action and context is not exclusive to video data. In this section, we perform some controlled experiments on still images. Action

and context components in still images can be easily annotated and visualized, so the experiments here are used for visualization and understanding of the learned network.

Dataset. The experiments in this section are performed on the PASCAL VOC-2012 Action dataset [5]. This dataset consists of 10 actions, *Jumping, Phoning, Playing Instrument, Reading, Riding Bike, Riding Horse, Running, Taking Photo, Using Computer, Walking*. There is also a distraction class *Others* where none of the aforementioned actions is performed. Each image contains one or multiple people with annotated bounding boxes and action labels. Note that each image is not exclusive to a single action, e.g., a person could be walking and phoning simultaneously.

Action and conjugate samples. Given an image of human action, we consider the action sample as the entire image (no bounding box information). To obtain the conjugate sample, we conceal all the human bounding boxes within that image using average pixel color. Thus the action and conjugate samples share the same background context and only differ in terms of the action. The human bounding boxes are only used when generating conjugate samples; they are not used in any other part of training and evaluation.

VGG-16 features. After training a VGG16 model [26], we extract the FC6 features to represent each action or conjugate sample. Given an image, following [26], we first resize it so that its smallest side equals 224, then densely apply the deep model on both the original and the horizontally flipped images to extract FC6 feature vectors. Subsequently, we perform average pooling and L_2 normalization. In the end, each image is represented by a 4096-dimensional vector.

Quantitative Evaluation. The baseline VGG16 [26] model achieves 74.1% mean AP on the validation set (no human bounding-box used). As expected, adding conjugate samples as either positive or negative training examples would degrade the performance, yielding mean AP of 73.1% and 72.6% respectively. Meanwhile, our factorization framework properly utilizes the conjugate samples and improves the classification performance to 75.2%.

Visualizing action and context. We visualize the image patches that excite the action and context extractors the most. For each image in the validation set, we divide it into 4×4 blocks and consider all 25 patches that correspond to 1×1 or 2×2 blocks. We feed each patch into our final network, and measure the action and context activation strengths separately. This can be effectively done as follows. To measure the excitement of the action extractor, we zero out the outputs of the context extractor g and record the value of the h classifier. This is taken as the excitement due to the action component in an image. Measuring the excitement of the context extractor can be done similarly. For each action class, we identify the image patches that have the highest action and context activation strengths. Figure 7 depicts some of these image patches for three action classes. Take *Jumping* as an example, the action extractor is most activated by the jumping poses, while the context extractor fires on lake scenes where jumping usually takes place.

Figure 7. **Representative patterns with highest action and context activation values.**

5. Summary and Discussion

We have proposed a method for separating human action from context without the need of detailed annotation. Our method is based on conjugate samples, which are training examples that are contextually similar to the action samples but do not contain the action. We performed experiments on several datasets and observed that: (1) our method for using conjugate samples to separate action from context led to improvement in the recognition performance; (2) there was qualitative and quantitative evidence that indicates some successes in separating action from context.

There are many scenarios where the separation of action and context are helpful, e.g., for transfer learning where the action classifier is used in a new domain with different context. There may also be a principled way to combine the action and context activation values to obtain a better classifier. These directions will be explored in our future work.

In this paper, the video sequences before and after the action sequences were used as conjugate samples, yielding excellent results. However, when these video sequences were not available due to the nature of existing data, we were forced to use "imperfect" conjugate samples. We still achieved significant performance gain, although less than the gain obtained using proper conjugate samples. As a future direction, we plan to develop a method to retrieve video clips that share similar background with action samples and use the retrieved clips as candidates for conjugate samples.

Acknowledgement. This project is supported by the National Science Foundation Award IIS-1566248 and a Google Research Award.

References

[1] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *Proceedings of the European Conference on Computer Vision*, 2014. 2

[2] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 2

[3] V. Delaitre, J. Sivic, and I. Laptev. Learning person-object interactions for action recognition in still images. In *Advances in Neural Information Processing Systems*, 2011. 2

[4] C. Desai and D. Ramanan. Detecting actions, poses, and objects with relational phraselets. In *Proceedings of the European Conference on Computer Vision*, 2012. 2

[5] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. www.pascal-network.org/challenges/VOC/voc2012/workshop/, 2012. 8

[6] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems*, 2016. 7

[7] R. Filipovych and E. Ribeiro. Recognizing primitive interactions by exploring actor-object states. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 2

[8] A. Gupta and L. Davis. Objects in action: An approach for combining action understanding and object perception. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[9] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009.

[10] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *Proceedings of the International Conference on Computer Vision*, 2009. 2

[11] M. Hoai and A. Zisserman. Thread-safe: Towards recognizing human actions across shot boundaries. In *Proceedings of the Asian Conference on Computer Vision*, 2014. 5

[12] M. Hoai and A. Zisserman. Improving human action recognition using score distribution and ranking. In *Proceedings of the Asian Conference on Computer Vision*, 2014. 2

[13] M. Hoai and A. Zisserman. Talking heads: Detecting humans and recognizing their interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2

[14] M. Hoai, Z.-Z. Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 2

[15] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *Proceedings of the European Conference on Computer Vision*, 2010. 2

[16] M. Jain, J. C. van Gemert, and C. G. M. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[17] M. Jiang, J. Kong, G. Bebis, and H. Huo. Informative joints based human action recognition using skeleton contexts. *Signal Processing: Image Communication*, 33:29–40, 2015. 2

[18] S. Jones and L. Shao. Unsupervised spectral dual assignment clustering of human actions in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2

[19] T. Lan, L. Sigal, and G. Mori. Social roles in hierarchical models for human activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 2

[20] L. J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In *Proceedings of the International Conference on Computer Vision*, 2007. 2

[21] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 2, 5, 7

[22] D. J. Moore, I. A. Essa, and M. H. H. III. Object spaces context management for human activity recognition. Technical Report GIT-GVU-98-26, Georgia Institute of Technology, 1998. 2

[23] A. Patron-Perez, M. Marszalek, A. Zisserman, and I. Reid. High five: Recognising human interactions in TV shows. In *Proceedings of British Machine Vision Conference*, 2010. 5

[24] V. Ramanathan, B. Yao, and L. Fei-Fei. Social role discovery in human events. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 2

[25] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014. 5

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015. 4, 8

[27] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human action classes from videos in the wild. Technical Report CRCV-TR-12-01, University of Central Florida, 2012. 7

[28] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 2

[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the International Conference on Computer Vision*, 2015. 3, 4, 5, 6, 7

[30] M. Ullah, S. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *Proceedings of the British Machine Vision Conference*, 2010. 2

[31] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 2

[32] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the International Conference on Computer Vision*, 2013. 5, 6, 7

[33] J. Wang, Z. Chen, and Y. Wu. Action recognition with mul-

tiscale spatio-temporal contexts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 2

[34] L. Wang, Y. Qiao, and X. Tang. Video action detection with relational dynamic-poselets. In *Proceedings of the European Conference on Computer Vision*, 2014. 2

[35] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *Proceedings of the European Conference on Computer Vision*, 2016. 7

[36] Y. Wang and M. Hoai. Improving human action recognition by non-action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 6

[37] Y. Wang, V. Tran, and M. Hoai. Eigen evolution pooling for human action recognition. *arXiv preprint arXiv:1708.05465*, 2017. 5, 7

[38] Y. Wang, V. Tran, and M. Hoai. Eigen-evolution dense trajectory descriptors. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2018. 5

[39] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg. A scalable approach to activity recognition based on object use. In *Proceedings of the International Conference on Computer Vision*, 2007. 2

[40] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2

[41] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2

[42] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2

[43] B. Yao, B. Nie, Z. Liu, and S. Zhu. Animated pose templates for modeling and detecting human actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3): 436–452, 2014. 2

[44] Y. Zhang, W. Qu, and D. Wang. Action-scene model for human action recognition from videos. In *AASRI Conference on Computational Intelligence and Bioinformatics*, 2014. 2