Multimodal Keyless Attention Fusion for Video Classification

Xiang Long[1], Chuang Gan[1]*, Gerard de Melo[2] , Xiao Liu[3] , Yandong Li[3] , Fu Li[3] , Shilei Wen[3]
[1]Tsinghua University , [2]Rutgers University , [3]Baidu IDL
{longx13, ganc13}@mails.tsinghua.edu.cn, gdm@demelo.org, {liuxiao12, liyandong, lifu, wenshilei}@baidu.com

Abstract

The problem of video classification is inherently sequential and multimodal, and deep neural models hence need to capture and aggregate the most pertinent signals for a given input video. We propose Keyless Attention as an elegant and efficient means to more effectively account for the sequential nature of the data. Moreover, comparing a variety of multimodal fusion methods, we find that Multimodal Keyless Attention Fusion is the most successful at discerning interactions between modalities. We experiment on four highly heterogeneous datasets, UCF101, ActivityNet, Kinetics, and YouTube-8M to validate our conclusion, and show that our approach achieves highly competitive results. Especially on large-scale data, our method has great advantages in efficiency and performance. Most remarkably, our best single model can achieve 77.0% in terms of the top-1 accuracy and 93.2% in terms of the top-5 accuracy on the Kinetics validation set, and achieve 82.2% in terms of GAP@20 on the official YouTube-8M test set.

Introduction

Video understanding is one of the most natural fundamental human abilities. From an early age, infants begin to recognize and understand events based on static imagery, motion, as well as sound. Although video classification is a very active research area in computer vision as well as machine learning, the current state-of-the-art remains subpar in comparison with human performance.

Unlike image classification, which takes still pictures as input, video classification is inherently multimodal, and thus image, motion, as well as sound cues may be necessary to make a comprehensive judgment. For instance, two musical instruments may be difficult to distinguish based on their mere appearance in a video, but might produce rather distinct sounds. In this case, acoustic features can greatly improve the accuracy of a video classification model.

As in other areas of computer vision, approaches based on deep convolutional neural networks (CNNs) have achieved state-of-the-art results. However, the improvements brought by CNNs, given their focus on local patterns, have not been as pronounced as for images. Due to the temporal sequential nature of videos, which can be very long, recurrent networks (RNNs) may be invoked to better capture longer-range temporal patterns and relationships. However, existing end-to-end approaches are restricted to small-scale datasets. It remains very difficult to combine CNN and RNN modeling for joint end-to-end training directly on large-scale datasets such as Kinetics and YouTube-8M.

A simpler approach to address this problem is to use a pre-trained model, or to train single-modality CNN models separately. Multimodal features can then more easily be extracted from a new video using the trained models. This has several crucial advantages: first, this facilitates transfer learning, such as from image classification and audio classification to video classification, or from one dataset to another. Additionally, the extracted features are significantly smaller in size than the raw RGB frame and audio

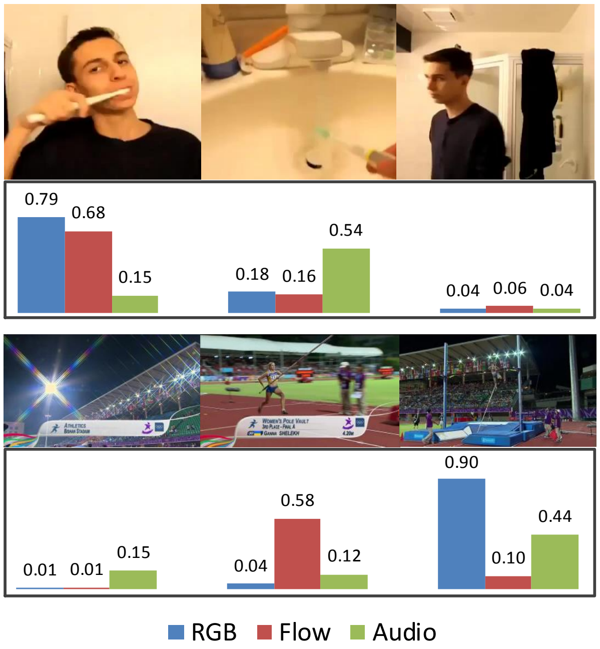

Figure 1: Examples of attention weights and corresponding temporal segments, where different modalities may focus on different time periods.

[*]Corresponding author.

data. Finally, large-scale datasets, such as YouTube-8M, often already provide pre-extracted features in advance, which greatly facilitates and accelerates research on such data.

Generally, each feature thus extracted stems from a particular local time segment in the video. The local features in chronological order constitute a complete feature sequence describing the video. Based on these, we can use a recurrent model to predict the output class distribution. We propose Keyless Attention as an elegant and efficient attention mechanism to achieve this more effectively and efficiently.

A second important observation is that we can represent a video more adequately by extracting *several* multimodal feature sequences. While we could rely on multiple separate RNN models for these different feature sequences, it is nontrivial to connect them, as one cannot simply piece together the features or use a simple ensemble.

Figure 1 provides examples of images for different time segments within two videos, showing how the importance of each modality varies across time. For the top example, during the brushing process, the image and motion is the clearest, and the RGB and flow features are most significant. In terms of audio, the sound of brushing is apparently insufficiently obvious, and the contribution of the audio signal is instead greatest when washing the toothbrush. For the bottom example, we encounter the greatest weight of motion features while the person is running, whereas the greatest weights for RGB and audio features are observed during the jump and landing phase. Hence, we conclude that different modalities may be pertinent at different time periods. Yet, there are also significant correlations between image and action (top), or image and sound (bottom), such that different modalities cannot be considered independently. Therefore, where and how to fuse multimodal RNN networks shall be examined in great detail in this paper.

Overall, we can summarize the main contributions of this paper as follows:

- We propose a simple and efficient attention mechanism that effectively aids the training of the RNN model.
- We analyze and study a variety of fusion methods for multimodal RNN-based architectures, and find that our proposed attention-based fusion robustly achieves the best results.
- We show that our proposed architecture performs robustly across four highly heterogeneous video classification datasets, including datasets with both trimmed and long untrimmed videos, and single-label as well as multi-label classification settings. We achieve highly competitive results on both the standard UCF-101 and ActivityNet datasets, as well as the challenging new Kinetics and YouTube-8M competitions, for which the release of the official results is still pending.

Related Work

Video Classification

Given the success that CNNs have enjoyed for image classification (Krizhevsky, Sutskever, and Hinton 2012; Szegedy et al. 2015; Simonyan and Zisserman 2014a; He et al. 2016), they have also been applied to the task of video classification (Karpathy et al. 2014; Gan et al. 2015; Simonyan and Zisserman 2014b; Gan et al. 2016b; Tran et al. 2015; Gan et al. 2016a; Varol, Laptev, and Schmid 2017; Carreira and Zisserman 2017). Initially, 2D CNNs were directly applied to RGB frames of videos. Karpathy et al. studied multiple approaches to extend the connectivity of a CNN across the time dimension to take advantage of local spatio-temporal information by pooling using single, late, early, or slow fusion (Karpathy et al. 2014). However, simple pooling does not bring significant gains compared to the single frame baseline.

To overcome the shortcomings of 2D CNNs and better account for spatio-temporal information, the optical flow method (Zach, Pock, and Bischof 2007) was proposed to consider the variation in the surrounding frames. Simonyan et al. proposed a method that uses RGB and stacked optical flow frames as appearance and motion signals, respectively (Simonyan and Zisserman 2014b). They show that the accuracy of action recognition is significantly boosted even by simply aggregating probability scores, which indicates that optical flow provides high-quality motion information. However, due to the inherent limitations of the optical flow method, it can only capture temporally local information.

Another method to obtain motion information is C3D (Tran et al. 2015), which is a natural extension of 2D CNNs. C3D relies on 3D convolution kernels to capture spatio-temporal information. Varol et al. found that expanding the temporal length of inputs for 3D CNNs can achieve better results and that using optical flows as inputs can outperform RGB inputs (Varol, Laptev, and Schmid 2017). Carreira et al. incorporated the Inception architecture (Szegedy et al. 2015) into 3D CNNs (Carreira and Zisserman 2017).

Many methods based on two-stream CNNs have been proposed to improve the accuracy of action recognition. Feichtenhofer et al. studied a number of ways of fusing CNNs both spatially and temporally in order to best take advantage of this spatio-temporal information (Feichtenhofer, Pinz, and Zisserman 2016). They also combined two-stream CNNs with ResNets (He et al. 2016) to show that the ResNet architecture is effective for action recognition with 2D CNNs (Feichtenhofer, Pinz, and Wildes 2016).

CNN methods excel at capturing short-term patterns in short, fixed-length videos, but it remains difficult to directly capture long-term interactions in long variable-length videos. Recurrent neural networks, particularly long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) ones, have been considered to model long-term temporal interactions in video classification. Ng et al. proposed two-stream LSTMs for higher-accuracy video classification over longer time periods (Ng et al. 2015). Donahue et al. devised an end-to-end architecture based on LSTMs for video classification and captioning (Donahue et al. 2015). Srivastava et al. first fine-tuned an LSTM in an unsupervised manner by using an encoder LSTM to map an input sequence to a fixed-length representation and then decoding it to reconstruct the input sequence or to predict the future sequence (Srivastava, Mansimov, and Salakhutdinov 2015). Subsequently, they adapted this pre-trained LSTM to video classification tasks. However, the accuracy and the training efficiency of RNN-based methods for video classification has been unsatisfactory, and how to fuse multimodal RNNs has not been studied in sufficient depth.

Attention Mechanisms

Humans recognize objects and events not by processing an entire visual scene simultaneously, but by selectively focusing on parts of the scene that provide the most pertinent information. Attention models were first proposed for object recognition with recurrent neural networks, drawing on the REINFORCE algorithm. In particular, Mnih et al. applied attention to extract information from an image by adaptively selecting a sequence of regions and sequentially considering the selected regions at a higher resolution (Mnih et al. 2014). Ba et al. presented an attention-based model to find the most relevant regions for recognizing multiple objects within images (Ba, Mnih, and Kavukcuoglu 2014).

Soft attention models were proposed to cope with tasks in natural language processing and visual captioning. In particular, Bahdanau et al. aimed at automatically capturing soft alignments between source words and target words in machine translation (Bahdanau, Cho, and Bengio 2014).

Subsequently, this soft attention model was applied to video classification tasks. Sharma et al. proposed a Soft-Attention LSTM model based on multi-layered RNNs to selectively focus on parts of the video frames and classify videos after taking a few glimpses (Sharma, Kiros, and Salakhutdinov 2015). Li et al. proposed an end-to-end sequence learning model called VideoLSTM (Li et al. 2016), which hardwires convolutions in the Soft-Attention LSTM. It stacks another RNN for motion modeling to better guide the attention towards the relevant spatial-temporal regions. However, these complex attention architectures are highly integrated with RNNs, and need to recalculate the attention weight map at every iteration. The attention modeling thus adds a significant burden to the computation and does not bring sufficient improvements in accuracy.

Multimodal Representation

Video is an inherently multimodal medium, with both visual and sound modalities. The video signal, moreover, can further be decomposed into static frames on the one hand, and signals capturing continuous motion on the other. Hence, using features of a single modality is clearly inadequate. In this paper, we use the following multimodal features to more thoroughly represent the contents of a video.

Visual Features

We rely on two kinds of visual features, RGB and flow features. RGB features are extracted directly from RGB images, while flow features are based on optical flow images created by stacking the two channels for the horizontal and vertical vector fields (Simonyan and Zisserman 2014b). We rely on deep convolution networks to extract these features. The models are initialized with pre-trained model from ImageNet and fine-tuned using the training split of the corresponding target dataset based on the temporal segment network framework (Wang et al. 2016a). After training, we can extract frame-level RGB and flow features for every frame in the video.

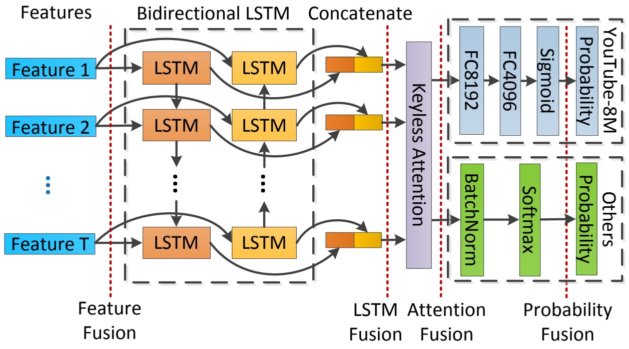

Figure 2: Our video classification model architecture based on Keyless Attention, where the red dotted lines represent four integration points corresponding to different multimodal fusion methods examined in this paper. For a specific integration point, we duplicate the network before the integration point K times for K different modalities, concatenate the variables at the integration point, and the network after the integration point remains unchanged.

Acoustic Features

We also use deep convolutional networks to extract acoustic features by preprocessing the raw audio to emit a sequence of matrices. The audio is first divided into non-overlapping 960ms frames. The frames are then decomposed with a short-time Fourier transform every 10ms and then aggregated, logarithm-transformed, into 64 mel-spaced frequency bins following (Hershey et al. 2016). This yields log-mel 96×64 spectrogram patches, which can be regarded as images. After obtaining spectrogram patches, we can extract frame-level acoustic features just as for visual features.

Segment-Level Features

Despite having obtained frame-level visual and acoustic features, we do not simply feed these into a recurrent network. First, the number of frames in a video can range to several thousands, which makes a direct application of LSTMs very challenging, as even LSTMs often fail to capture particularly long-range dependencies. Second, the features of successive frames tend to be very similar, and it is not necessary to input all of them into the network. Third, a large number of frame-level features can require too much memory and make the training process both slower and more difficult.

We hence use temporal adaptive pooling to obtain segment-level features for both visual and acoustic signals. Specifically, we use 1D adaptation max-pooling to transform all the frame-level features into an equal number of segment-level features, such that each segment-level feature corresponds to roughly equal-length temporal segments of the video, and each modality of a given video has the same number of features.

Proposed Network Architecture

In this section, we describe our proposed approach of Multimodal Keyless Attention Fusion for video classification. Figure 2 illustrates the architecture of our model. We first introduce a novel keyless attention mechanism, and then describe attention-based multimodal fusion in detail.

Keyless Attention

Our first contribution is a simple, efficient, and effective attention mechanism for video classification. Given a sequence of input vectors $\{a_1, a_2, \ldots, a_n\}$, which we call *annotation vectors*, this attention mechanism seeks to compute an output vector c given by taking the expectation over the annotation vectors:

$$c = \sum_{i=1}^{n} \lambda_i a_i \tag{1}$$

The weight of each a_i is computed by:

$$e_i = w^T a_i \tag{2}$$

$$\lambda_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)} \tag{3}$$

where w is a learnable parameter of the same dimensionality as the annotation vectors. For convenience, we denote the keyless attention model's output as $c = \mathrm{KeylessAtt}(\{a_i\})$.

Existing soft attention mechanisms (Bahdanau, Cho, and Bengio 2014) compute the weight not only based on the annotation vectors, but rely on an additional input, such as the previous hidden state vector of the LSTM, or a vector representing some target entity (Wang et al. 2016b). We refer to such vectors as *key vectors*, because soft attention essentially seeks to find the most related weighted average of annotations according to this key vector, and different key vectors will result in different weighted averages. Unlike soft attention mechanisms, the weights in our approach only depend on the annotations (which will be the outputs of the bidirectional LSTM in this paper). Because our attention mechanism does not rely on key vectors as input, we refer to it as Keyless Attention.

Therefore, the attention vector only needs to be computed once for each video, rather than at every time step in the RNN. The operation to compute the attention weights in this keyless attention model is essentially a 1D-convolutional operation with 1 filter, which is very efficient both in time and space. We have also evaluated other methods to compute the attention weights, but found that more complex architectures do not lead to improvements. Hence, we pick this simple and elegant model structure.

Recurrent Video Classification Model based on Keyless Attention

Our architecture build upon a bidirectional LSTM (Hochreiter and Schmidhuber 1997) as shown in Figure 2. Given an input feature sequence $(x_1, x_2, \ldots, x_T)$, an LSTM unit computes the hidden state $(h_1, h_2, \ldots, h_T)$ via repeated application of the following equations:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{5}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{6}$$

$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \tag{7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{8}$$

$$h_t = o_t \odot c_t, \tag{9}$$

where $\sigma(\cdot)$ is the sigmoid function and $\odot$ denotes element-wise multiplication. Bidirectional LSTMs compute two separate forward and backward passes yielding two sequences of hidden states, which we denote as h_t^{f} and h_t^{b}, respectively, at each time step t. We obtain the output at each time step of a bidirectional LSTM as $h_t^{\mathrm{B}} = [h_t^{\mathrm{f}}, h_t^{\mathrm{b}}]$, where $[\cdot]$ denotes the concatenation of the state vectors.

It is challenging to directly feed all these states $(h_1^{\mathrm{B}}, h_2^{\mathrm{B}}, \ldots, h_T^{\mathrm{B}})$ to the network for classification. This is because each video may be of a different length, resulting in heterogeneous dimensionalities, and because the length of the video may be large, leading to overly high input dimensionalities. We thus compute a fixed-dimensional global representation g. For this, we rely on our Keyless Attention mechanism:

$$g = \mathrm{KeylessAtt}(h_1^{\mathrm{B}}, h_2^{\mathrm{B}}, \ldots, h_T^{\mathrm{B}}) \tag{10}$$

Intuitively, this mechanism can be viewed as taking a global look at the video and quickly identifying the most-contributing features for the classification. We find that this leads to significant gains compared to using the average of states or the final state of bidirectional LSTMs.

After obtaining the global representation g, we apply several fully connected layers (FC) or a batch normalization layer (BN) (Ioffe and Szegedy 2015) to compute the probabilities for classes y_i for the i-th video.

For the single-label classification datasets (e.g., UCF101, ActivityNet, and Kinetics), we apply a BN layer followed by one FC classification layer and a softmax, and for multi-label classification (e.g., on YouTube-8M), we apply two sequential FC layers with 8192 / 4096 hidden units, respectively, and $\tanh$ activation, followed by a FC classification layer with sigmoid activation. Because the pre-extracted features provided for YouTube-8M have been preprocessed via PCA for dimensionality reduction followed by quantization, the variance between features is not large. We hence do not apply batch normalization to YouTube-8M.

We represent the ground truth classes for the i-th video as a one-hot vector $\hat{y}_i$ and the number of videos in the training set is denoted as N. For single-label classification, we can train the model by minimizing the loss function:

$$\frac{1}{N} \sum_{i=1}^{N} y_i^T \log(\hat{y}_i), \tag{11}$$

while for multi-label classification, we minimize the following loss function:

$$\frac{1}{N} \sum_{i=1}^{N} [y_i^T \log(\hat{y}_i) + (1 - y_i)^T \log(1 - \hat{y}_i)] \tag{12}$$

Multimodal Fusion

Previously, we have describe a video classification model for a single sequence of features. However, in order to fully exploit the multimodal nature of videos, we need to account for multiple modalities. Given multiple feature sequences as input, how shall we perform multimodal fusion to obtain the best-possible results? We assume we are given K different feature sequences $(x_1^{(1)}, ..., x_T^{(1)}), \ldots, (x_1^{(K)}, ..., x_T^{(K)})$. Note that these are segment-level features as described earlier, and that each feature sequence is guaranteed to be of the same length T.

We consider four different multimodal fusion methods corresponding to four different integration points (red dotted line) in Figure 2.

Feature Fusion: A simple feature-level fusion is one of the most intuitive methods. Each feature represents a particular temporal segment in the video. Stitching together the features of the same segment leads to a more detailed representation of that temporal segment. Hence, we can fuse these feature sequences into a single feature sequence $(x_1, \ldots, x_T)$ by applying $x_t = [x_t^{(1)}, \ldots, x_t^{(K)}]$ and we can perform video classification just as for a single feature sequence.

LSTM Fusion: This method is similar to feature-level fusion in that it also operates at the segment level. Each input feature sequence $(x_1^{(i)}, \ldots, x_T^{(i)})$ obtains its own hidden states $(h_1^{B(i)}, \ldots, h_T^{B(i)})$ via a separate bidirectional LSTM. Then we obtain the fused hidden states $(h_1^{\mathrm{B}}, h_2^{\mathrm{B}}, \ldots, h_T^{\mathrm{B}})$ by applying $h_t^{\mathrm{B}} = [h_t^{B(1)}, \ldots, h_t^{B(K)}]$. After this, we can compute the global representation and invoke the subsequent layers just as for a single feature sequence.

Attention Fusion: Unlike the previous two fusion methods, attention fusion operates at the level of videos. First, we obtain the global representation $g^{(i)}$ for each input sequence through a bidirectional LSTM and separate Keyless Attention models. Then, we fuse the global representations as $g = [g^{(1)}, \ldots, g^{(K)}]$ and predict the classes by invoking the subsequent layers. Note that this fusion method does not require that each feature sequence be of the same length.

Probability Fusion: Finally, Probability Fusion is essentially an ensemble method. We first train a separate model for each of the feature sequences just as when we only have a single feature sequence. We then average the outputs of K independent models as the final output. In this approach, each model can only access features of a single modality, and thus interactions across modalities cannot be learned.

Experimental Results

In this section, we proceed to evaluate and compare our proposed methods in terms of their classification accuracy.

Datasets

We evaluate our approach on four popular video classification datasets.

UCF101 (Soomro, Roshan Zamir, and Shah 2012) is a trimmed video dataset, consisting of realistic web videos with diverse forms of camera motion and illumination. It contains 13,320 video clips with an average length of 180 frames per clip. These are labeled with 101 action classes, ranging from daily life activities to unusual sports. Each video clip is assigned just a single class label. Following the original evaluation scheme, we report the average accuracy over three training/testing splits.

ActivityNet (Heilbron et al. 2015) is an untrimmed video dataset. We use the ActivityNet v1.3 release, which consists of more than 648 hours of untrimmed videos from a total of around 20K videos with 1.5 annotations per video, selected from 200 classes. Videos can contain more than one activity, and, typically, large time segments of a video are not related to any activity of interest. In the official split, the distribution among training, validation, and test data is about 50%, 25%, and 25% of the total videos, respectively. Because the annotations for the testing split have not yet been published, we report experimental results on the validation split.

Kinetics (Carreira and Zisserman 2017) is a trimmed video dataset. The dataset contains 246,535 training videos, 19,907 validation videos, and 38,685 test videos, covering 400 human action classes. Each clip lasts around 10s and is labeled with a single class. The annotations for the test split have not yet been released, so we report experimental results on the validation split.

YouTube-8M (Abu-El-Haija et al. 2016) is massively large untrimmed video dataset. It contains over 1.9 billion video frames and 8 million videos. Each video can be annotated with multiple tags. Visual and audio features have been pre-extracted and are provided with the dataset for each second of the video. The visual features were obtained via a Google Inception CNN pre-trained on ImageNet (Deng et al. 2009), followed by PCA-based compression into a 1024-dimensional vector. The audio features were extracted via a pre-trained VGG-inspired (Simonyan and Zisserman 2014a) network. In the official split, the distribution among training, validation, and test data is about 70%, 20%, and 10%, respectively. As the annotations of the test split have not been released to the public and the number of videos in the validation set is overly large, we maintain 60K videos from the official validation set to validate the parameters. Other videos in the validation set are included into the training set. We report experimental results on this 60K validation set and on the official test set.

Implementation Details

For UCF101 and ActivityNet, we extract both RGB and flow features using a ResNet-152 (He et al. 2016) model. For Kinetics, we extract RGB and flow features using Inception-ResNet-v2 (Szegedy et al. 2016) and extract audio features with a VGG-16 (Simonyan and Zisserman 2014a). The number of segments we used for fine-tuning is 3 for UCF101, and 7 for ActivityNet and Kinetics to strike a balance between recognition performance and computational burden.

We max-pool the frame-level features to 5 segment-level features for UCF101 and Kinetics, where the lengths of

Method		Accuracy(%)
iDT + FV (Wang and Schmid 2013)		85.9
iDT + HSV (Peng et al. 2016)		87.9
EMV-CNN (Zhang et al. 2016)		86.4
Two Stream (Simonyan and Zisserman 2014b)		88.0
FSTCN (Sun et al. 2015)		88.1
VideoLSTM(Li et al. 2016)		89.2
TDD+FV (Wang, Qiao, and Tang 2015)		90.3
Fusion (Feichtenhofer, Pinz, and Zisserman 2016)		92.5
TSN(3 seg) (Wang et al. 2016a)		94.2
ST-ResNet+iDT (Feichtenhofer, Pinz, and Wildes 2016)		94.6
ActionVLAD (Girdhar et al. 2017)		93.6
Ours	RGB CNN	85.4
	RGB Average	85.9
	RGB Last	85.8
	RGB Attention	**86.2**
	Flow CNN	86.2
	Flow Attention	**87.0**
	Feature Fusion	94.1
	LSTM Fusion	94.2
	Probility Fusion	93.5
	Attention Fusion	**94.8**

Table 1: Mean classification accuracy on UCF-101.

videos are a few seconds, and 20 for ActivityNet, where the video length is relatively large. Different from (Wang et al. 2016a), which used the trimmed videos for training, we train our model using the whole untrimmed video from ActivityNet to consider more realistic input conditions. For YouTube-8M, we directly use the provided pre-extracted RGB and audio features as segment-level features, where the maximum number of segment is 300.

The number of hidden units for the LSTM on UCF101, ActivityNet, and Kinetics is 512, which is a standard choice, while for YouTube-8M, we use 1024 to handle its longer videos.

We rely on the RMSPROP algorithm (Tieleman and Hinton 2012) to update parameters for better convergence, with a learning rate of 0.0001.

Results for Keyless Attention

In this section, we verify the effectiveness of the proposed Keyless Attention mechanism. We perform all experiments on a single RGB feature sequence, in order to eliminate any interference between modalities. Our proposed model with Keyless Attention (RGB Attention) is hence the model described in Section .

The results of the CNN model are reported as RGB CNN, Flow CNN, and Audio CNN in the tables. We also compare our method with two RNN-based classification methods:

The first method (RGB Last) is to concatenate the two final hidden states of a bidirectional LSTM, and then apply the subsequent layers for classification just as for the attention-based recurrent model. The second method (RGB Average) averages the outputs of the bidirectional LSTM and then classifies the averaged state vector.

The experimental results on each dataset in Tables 1–3 show that Keyless Attention outperforms both RGB Last and RGB Average. Moreover, during training, we also find that the use of Keyless Attention leads to a faster convergence speed, and virtually no additional compute time per batch, such that we can finish training in a shorter period of time. We conclude that the proposed RNN model based on Keyless Attention is not only simple, but also efficient and effective.

Method		mAP(%)
iDT + FV (Peng et al. 2016)		66.5
Depth2Action (Zhu and Newsam 2016)		78.1
Two Stream (Simonyan and Zisserman 2014b)		71.9
C3D (Tran et al. 2015)		74.1
Ours	RGB CNN	70.9
	RGB Average	71.3
	RGB Last	71.4
	RGB Attention	**72.0**
	Flow CNN	64.0
	Flow Attention	**64.8**
	Feature Fusion	77.5
	LSTM Fusion	77.6
	Probility Fusion	77.2
	Attention Fusion	**78.5**

Table 2: ActivityNet results on the validation set.

Results for Multimodal Fusion

In this section, we analyze the experimental results and discuss the characteristics of the four proposed multimodal fusion methods:

Probability Fusion: As this is a fusion method that can be regarded as a form of ensembling, each modality is completely independent of others, and the independently obtained results are fused in a final step. Since the different modalities are only combined at the final step of averaging the output probabilities from independent models, we are unable to learn any interactions between modalities. This explains the poor results achieved by this approach.

From the experimental results, as shown in Tables 1–4, we find that across all four datasets, the Probability Fusion approach performs worst among the four fusion methods. This difference is particularly pronounced on the two large-scale datasets Kinetics and YouTube-8M. On Kinetics, the difference in accuracy between Probability Fusion and the best result is 2.1%, and on YouTube-8M, the GAP@20 for Probability Fusion is 1.8% lower than for the best result. We hypothesize that, on large datasets, we have more data from which one can learn multimodal associations, leading to more pronounced gaps.

Feature Fusion: This is the most intuitive and easy to implement method. It directly connects multiple modalities within each local time interval. However, in this case, the attention will select periods of time globally, neglecting that there may be a need to focus on diverging periods of time for different modalities. Hence, the bidirectional LSTM may need to learn to align all the salient signals across modalities, while simultaneously also needing to learn interactions between different modalities and temporal relationships. This burden may be too heavy.

We can observe that the results obtained by this approach on the four datasets are better than those of Probability Fusion, but worse than the best result. This illustrates that the Feature Fusion method can learn part of the association between modalities. We also observe that the longer the sequence of features (such as for YouTube-8M, in which the length of the video can be 300 segments), the more obvi-

	Method	Top-1(%)	Top-5(%)
C3D (Tran et al. 2015)		55.6	79.1
3D ResNet (Hara, Kataoka, and Satoh 2017)		58.0	81.3
Two-Stream I3D* (Carreira and Zisserman 2017)		74.2	91.3
Ours	RGB CNN	73.0	90.9
	RGB Average	73.2	91.1
	RGB Last	73.0	91.0
	RGB Attention	**73.8**	**91.3**
	Flow CNN	54.5	75.9
	Flow Attention	**54.9**	**76.4**
	Audio CNN	21.6	39.4
	Audio Attention	**22.0**	**40.1**
	Feature Fusion	76.1	92.6
	LSTM Fusion	76.2	92.6
	Probility Fusion	74.9	91.6
	Attention Fusion	**77.0**	**93.2**

Table 3: Kinetics results on the validation set, except for those marked with '$*$', which are based on the test set.

ous the performance drop. We conjecture that on longer sequences, the LSTM will need to expend more effort in learning temporal connections rather than focusing on modal interactions.

LSTM Fusion: This method is similar to Feature Fusion in that the burden of learning associations across modalities and of temporal connections within individual modalities still falls on the LSTM.

The experimental results obtained by this method are almost the same as for Feature Fusion, or in some cases slightly better, which is in line with our expectations.

Attention Fusion: Unlike Feature Fusion and LSTM Fusion, this approach does not force the features of different modalities within the same time period to be linked. Hence, the LSTM can focus on uncovering temporal patterns within individual modalities and for each modality can effortlessly pay attention to different temporal segments. However, the Attention Fusion method is different from Probability Fusion in that the network still has the opportunity to capture interactions between modalities.

Figure 1 provides an example of attention weights for images in the corresponding temporal segment. We observe that RGB, optical flow, and audio signals can independently attend to different areas of interest.

Considering the experimental results, we find that the Attention Fusion method is the most effective across all of the four datasets. This fusion method can take into account both the temporal progression and multimodal interactions, which shows to be the best way of fusing attention-based recurrent components. We refer to this method as Multimodal Keyless Attention Fusion.

Comparison with State-of-the-Art

Finally, we compare our method against the state-of-the-art methods. On UCF101 and ActivityNet, we compare it with some of the existing published traditional methods (Wang and Schmid 2013; Peng et al. 2016) as well as deep learning approaches (Zhang et al. 2016; Simonyan and Zisserman 2014b; Sun et al. 2015; Li et al. 2016; Wang, Qiao, and Tang 2015; Feichtenhofer, Pinz, and Zisserman 2016; Tran et al. 2015; Wang et al. 2016a; Feichtenhofer, Pinz, and Wildes 2016; Girdhar et al. 2017). Our approach can achieve competitive results.

	Method	60K Valid(%)	Test(%)
VLAD (Xu, Yang, and Hauptmann 2015)		-	80.4
Video Level (Zhong et al. 2017)		-	78.6
LSTM + MoE (Wang et al. 2017)		-	80.2
Ours	RGB Attention	76.6	77.3
	Audio Attention	54.0	-
	Feature Fusion	80.5	81.5
	Probility Fusion	79.1	-
	Attention Fusion	**80.9**	**82.2**

Table 4: YouTube-8M GAP@20 on the 60K validation and test set.

On Kinetics, we compare it with recently published results (Tran et al. 2015; Hara, Kataoka, and Satoh 2017; Carreira and Zisserman 2017). Note that result of (Carreira and Zisserman 2017) was reported based on the test set, for which the top-1 results empirically are 1.5% lower than the top-1 on the validation set. On YouTube-8M, we compare it with VLAD (Jegou et al. 2010; Xu, Yang, and Hauptmann 2015) and recently published results (Zhong et al. 2017; Wang et al. 2017). Our approach enables our team to achieve very strong results in both the Kinetics and YouTube-8M competitions. To facilitate further comparison, Table 4 also provides the single model results on the official YouTube-8M test set, as computed by the competition organizers.

Unlike these recently published works, our model robustly achieves competitive results across a range of different datasets, including smaller and larger training sets, and trimmed and untrimmed videos.

Conclusion

To better cope with the sequential and multimodal nature of videos, we have proposed a keyless attention mechanism, which allows for fast and effective learning of RNN models. We further assess several alternatives for achieving multimodal fusion with recurrent neural networks, and find that the proposed attention-based fusion achieves the best results. We have conducted experiments on four well-known datasets, including both untrimmed and trimmed videos, single-label and multi-label classification settings, and small-scale as well as very large-scale datasets. Our highly competitive results in all of these settings demonstrate that our proposed Multimodal Keyless Attention Fusion framework is robust across a large range of video classification tasks.

In terms of future work, we hope to apply this fusion approach in end-to-end-trained CNN and RNN architectures for further gains.

Acknowledgments

Gerard de Melo's research is supported in part by the DARPA SocialSim program.

References

Abu-El-Haija, S.; Kothari, N.; Lee, J.; Natsev, P.; Toderici, G.; Varadarajan, B.; and Vijayanarasimhan, S. 2016. YouTube-8M: A Large-Scale Video Classification Benchmark. *ArXiv*.

Ba, J.; Mnih, V.; and Kavukcuoglu, K. 2014. Multiple object recognition with visual attention. *arXiv:1412.7755*.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*.

Carreira, J., and Zisserman, A. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. *ArXiv*.

Deng, J.; Dong, W.; Socher, R.; and Li, L. J. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Donahue, J.; Hendricks, L. A.; Rohrbach, M.; Venugopalan, S.; Guadarrama, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.

Feichtenhofer, C.; Pinz, A.; and Wildes, R. P. 2016. Spatiotemporal residual networks for video action recognition. In *NIPS*.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional two-stream network fusion for video action recognition. In *CVPR*.

Gan, C.; Wang, N.; Yang, Y.; Yeung, D.-Y.; and Hauptmann, A. G. 2015. Devnet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*, 2568–2577.

Gan, C.; Sun, C.; Duan, L.; and Gong, B. 2016a. Webly-supervised video recognition by mutually voting for relevant web images and web video frames. In *ECCV*, 849–866.

Gan, C.; Yao, T.; Yang, K.; Yang, Y.; and Mei, T. 2016b. You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images. In *CVPR*, 923–932.

Girdhar, R.; Ramanan, D.; Gupta, A.; Sivic, J.; and Russell, B. 2017. ActionVLAD: Learning spatio-temporal aggregation for action classification. In *CVPR*.

Hara, K.; Kataoka, H.; and Satoh, Y. 2017. Learning Spatio-Temporal Features with 3D Residual Networks for Action Recognition. *ArXiv*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Heilbron, F. C.; Escorcia, V.; Ghanem, B.; and Niebles, J. C. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*.

Hershey, S.; Chaudhuri, S.; Ellis, D. P. W.; Gemmeke, J. F.; Jansen, A.; Channing Moore, R.; Plakal, M.; Platt, D.; Saurous, R. A.; Seybold, B.; Slaney, M.; Weiss, R. J.; and Wilson, K. 2016. CNN Architectures for Large-Scale Audio Classification. *ArXiv*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Ioffe, S., and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ArXiv*.

Jegou, H.; Douze, M.; Schmid, C.; and Perez, P. 2010. Aggregating local descriptors into a compact image representation. In *CVPR*.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Li, F. F. 2014. Large-scale video classification with convolutional neural networks. In *CVPR*, 1725–1732.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. In *NIPS*.

Li, Z.; Gavves, E.; Jain, M.; and Snoek, C. G. M. 2016. VideoLSTM Convolves, Attends and Flows for Action Recognition. *ArXiv*.

Mnih, V.; Heess, N.; Graves, A.; et al. 2014. Recurrent models of visual attention. In *NIPS*, 2204–2212.

Ng, Y. H.; Hausknecht, M.; Vijayanarasimhan, S.; Vinyals, O.; Monga, R.; and Toderici, G. 2015. Beyond short snippets: Deep networks for video classification. In *CVPR*.

Peng, X.; Wang, L.; Wang, X.; and Qiao, Y. 2016. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *CVIU* 150(C):109–125.

Sharma, S.; Kiros, R.; and Salakhutdinov, R. 2015. Action Recognition using Visual Attention. *ArXiv*.

Simonyan, K., and Zisserman, A. 2014a. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv*.

Simonyan, K., and Zisserman, A. 2014b. Two-stream convolutional networks for action recognition in videos. *NIPS*.

Soomro, K.; Roshan Zamir, A.; and Shah, M. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *ArXiv*.

Srivastava, N.; Mansimov, E.; and Salakhutdinov, R. 2015. Unsupervised learning of video representations using LSTMs. In *ICML*.

Sun, L.; Jia, K.; Yeung, D. Y.; and Shi, B. E. 2015. Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*, 1–9.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *ArXiv*.

Tieleman, T., and Hinton, G. 2012. Lecture 6.5: RMSprop. *Coursera: Neural Networks for Machine Learning*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

Varol, G.; Laptev, I.; and Schmid, C. 2017. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP(99):1–1.

Wang, H., and Schmid, C. 2013. Action recognition with improved trajectories. In *ICCV*.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Gool, L. V. 2016a. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*.

Wang, L.; Cao, Z.; de Melo, G.; and Liu, Z. 2016b. Relation classification via multi-level attention CNNs. In *ACL*.

Wang, Z.; Kuan, K.; Ravaut, M.; Manek, G.; Song, S.; Fang, Y.; Kim, S.; Chen, N.; D'Haro, L. F.; Tuan, L. A.; Zhu, H.; Zeng, Z.; Cheung, N. M.; Piliouras, G.; Lin, J.; and Chandrasekhar, V. 2017. Truly Multi-modal YouTube-8M Video Classification with Video, Audio, and Text. *ArXiv*.

Wang, L.; Qiao, Y.; and Tang, X. 2015. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*.

Xu, Z.; Yang, Y.; and Hauptmann, A. G. 2015. A discriminative CNN video representation for event detection. In *CVPR*.

Zach, C.; Pock, T.; and Bischof, H. 2007. A duality based approach for realtime tv-l1 optical flow. *DAGM* 4713(5):214–223.

Zhang, B.; Wang, L.; Wang, Z.; Qiao, Y.; and Wang, H. 2016. Real-time action recognition with enhanced motion vector CNNs. In *CVPR*.

Zhong, Z.; Huang, S.; Zhan, C.; Zhang, L.; Xiao, Z.; Wang, C.-C.; and Yang, P. 2017. An Effective Way to Improve YouTube-8M Classification Accuracy in Google Cloud Platform. *ArXiv*.

Zhu, Y., and Newsam, S. 2016. Depth2action: Exploring embedded depth for large-scale action recognition. In *ECCV*, 668–684.