Stochastic Fusion for Multi-stream Neural Network in Video Classification

Yu-Min Huang* Huan-Hsin Tseng† Jen-Tzung Chien*
* Department of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan
† Department of Radiation Oncology, University of Michigan, Ann Arbor, MI, USA
E-mail: patrick.eed03@g2.nctu.edu.tw; thuanhsi@umich.edu, jtchien@nctu.edu.tw

Abstract—**Spatial image and optical flow provide complementary information for video representation and classification. Traditional methods separately encode two stream signals and then fuse them at the end of streams. This paper presents a new multi-stream recurrent neural network where streams are tightly coupled at each time step. Importantly, we propose a stochastic fusion mechanism for multiple streams of video data based on the Gumbel samples to increase the prediction power. A stochastic backpropagation algorithm is implemented to carry out a multi-stream neural network with stochastic fusion based on a joint optimization of convolutional encoder and recurrent decoder. Experiments on UCF101 dataset illustrate the merits of the proposed stochastic fusion in recurrent neural network in terms of interpretation and classification performance.**

I. Introduction

In recent years, deep learning has achieved a great success in different emerging tasks and challenging domains where a variety of information systems in computer vision and natural language processing have been constructed. The key to this success is because deep neural networks can capture the complicated high-dimensional mapping between input data and output targets. As we know, the convolutional neural network (CNN) [1] and the recurrent neural network (RNN) [2][3] are two representative paradigms in deep learning which are powerful to learn different kinds of spatial and temporal data. In general, CNN is fitted to various computer vision tasks because the convolution layers are suitable to extract the spatial features from an image while the max-pooling layers make the trained model robust to noise interference and invariant to feature shifting. CNN has been widely extending to build numerous complex systems with state-of-the-art performance. On the other hand, RNN is seen as the specialized neural architecture with recurrent feedback which is suitable to characterize the sequential patterns and reflect the temporal behaviors via the cell or internal memory based on a dynamic state over time. Apparently, video data are regarded as the sequential signals which contain both spatial and temporal information. Intuitively, CNN and RNN can be applied for video representation and classification. A simple idea is to utilize CNN to encode the images and then apply RNN to represent the causal relations among frames.

For video classification, some papers aim to introduce hand-crafted features into deep learning framework. The most popular hand-crafted feature is optical flow. Optical flow [4] can be viewed as a set of displacement vector fields between the pairs of consecutive frames. Empirically, trying to capture temporal information with single stream is not enough since optical flow of images provides additional information for video representation. Therefore, two-stream based approaches [5] are mainstream in video classification nowadays. One stream is original frames, which is also known as spatial stream. The other is optical flow. Because optical flow is about movement between consecutive frames, we can see it as temporal stream. The experiment shows that with the help of optical flow, the performance can be improved significantly.

A crucial issue in video classification is to design an effective approach to encode two streams and then fuse those streamed features to identify the corresponding classes. Traditionally, this issue was tackled by separately encoding two streams for final fusion of their hidden codes in the end of video clip [5]. The mutual information of two streams at each individual time step was totally disregarded. The classification performance was constrained. This paper presents a stochastic multi-stream network where CNN encoder is applied to extract two streams of features from raw images and optical flows and then RNN decoder with stochastic fusion at each time frame is performed for video classification. Importantly, the hidden states of two streams are considered at each time. The mutual information between two streams is characterized. With a reasonable stochastic fusion, this multi-stream attention network can improve the classification performance when compared with the single stream network, the separate multi-stream network and the single combined stream networks. Visualization of stream weights in time axis can interpret the tradeoff between spatial images and optical flows in a video clip. Our contribution can be summarized as follows:

1) We introduce stochastic fusion on hidden states of LSTMs, which makes information flows mix more reasonably, and gains benefits from mutual information.
2) Taking advantage of better information fusion, our model outperforms other RNN based methods.
3) Because of stochastic fusion, the usage of multi-stream is more interpretable and comprehensive now.

In next section, we will introduce some related works and how others dealt with this issue recently.

II. Background Survey

This paper deals with the video classification for different actions where a sequence of frames in a video clip is observed.

Each static raw image depicts the spatial content at each time step. In addition to the spatial resolution manifested by a raw image at each time step, the optical flow [4] provides a complementary information besides visual representation. Relative to the spatial resolution from a single image frame, the temporal evidence based on the optical flow is calculated from consecutive frames which are collected at each time step. In particular, the optical flow by definition measures the displacement vector fields between pairs of consecutive frames. Such measurement provides instantaneous temporal dynamics at each time instant. In practice, the displacement vector fields can be further decomposed into horizonal and vertial directions according to the Euclidean coordinates, denoted as optical x and y respectively. Consequently, one has *three* aligned and complementary streams which can be combined to improve the accuracy for video classification. Without loss of generality, in the following discussion we simply formulate our solution as two-stream data, where in general our method can be straightforwardly extend to multiple streams [6].

A. Information Fusion

A direct solution to characterize the long-term temporal information for stream data is based on the RNN or long short-term memory (LSTM) [7]. To use the two streams of video data, [8] combined the stream data at the embedding or encoding stage with CNNs and a single LSTM was then applied for video classification. More meaningfully, two data streams can be separately encoded by CNNs and individually processed by two different LSTMs. The optical flow reveals the movement of consecutive frames of temporal instants. Given the spatial stream and optical flow, two LSTMs can be separately run and jointly fused at the last time frame T [5], where this separate multi-stream network would perform better than the single spatial stream network for action recognition. However, the information integration can also be executed by the linear transformation and combination [9] or via the adaptive weight fusion [10]. The fused features was treated as a context vector for classification based on the support vector machine [11]. Overall, the movement in a video clip is highly correlated to the spatial scene, and thus treating the streams independently may suffer from the loss of video content. It is always encouraged to combine two streams early before the final stage. This study works toward refining the video task by synchronously fusing and exchanging different stream sources at each time step using a recurrent machine.

B. Stream Interpretation

In fact, the fusion over different streams of features or embeddings can be interpreted in two ways with different perspectives. First, the fusion over steams is regarded as an *attention* over time-dependent latent codes in many temporal deep learning algorithms. The solutions to attention mechanism could be helpful for information fusion. In [12], the transformer was proposed to realize the attention mechanism [13] so as to achieve the claim that all the need for sequential learning was attention. In [14], the attention method was introduced to fuse the multimodal features or hidden states over different time steps as a global view of the whole video. Attention was applied over time while we concern about the fusion over steams. On the other hand, a recent work, called the Markov RNN [15], [16], was proposed to characterize multiple *states* behind an observed sequence data based a stochastic Markov process. The scheme to indicate the discrete hidden state or identify the one-hot vector $\mathbf{z_t}$ at each time frame $\mathbf{x_t}$ can be adopted as a fusion mechanism to combine multiple streams if the sequences of stream data are observed. This paper is motivated to carry out a fusion mechanism over the state space based on a multi-stream RNN framework. A streamed LSTM is implemented in an efficient way.

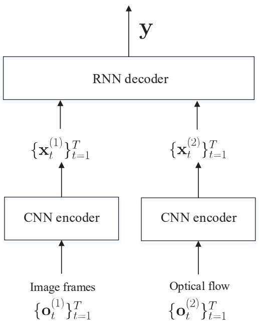

$\mathbf{y}$
RNN decoder
$\{\mathbf{x}_t^{(1)}\}_{t=1}^T$
$\{\mathbf{x}_t^{(2)}\}_{t=1}^T$
CNN encoder
CNN encoder
Image frames
$\{\mathbf{o}_t^{(1)}\}_{t=1}^T$
Optical flow
$\{\mathbf{o}_t^{(2)}\}_{t=1}^T$

Fig. 1. Multi-stream encoders and decoder for classification output.

III. Multi-stream Fusion Network

A multi-stream fusion network is built by the CNN stream encoder and the RNN fusion decoder with a stochastic fusion which are jointly trained by the backpropagation algorithm. Figure 1 shows the construction of CNN encoder and RNN decoder for finding classification output $\mathbf{y}$ of a video clip with the detailed descriptions provided below.

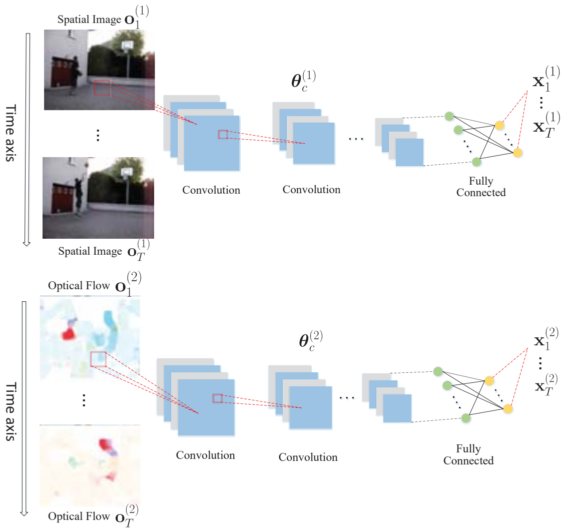

Spatial Image $\mathbf{o}_1^{(1)}$
Time axis
$\boldsymbol{\theta}_c^{(1)}$
$\mathbf{x}_1^{(1)}$
$\mathbf{x}_T^{(1)}$
Convolution
Convolution
Fully Connected
Spatial Image $\mathbf{o}_T^{(1)}$
Optical Flow $\mathbf{o}_1^{(2)}$
Time axis
$\boldsymbol{\theta}_c^{(2)}$
$\mathbf{x}_1^{(2)}$
$\mathbf{x}_T^{(2)}$
Convolution
Convolution
Fully Connected
Optical Flow $\mathbf{o}_T^{(2)}$

Fig. 2. CNN encoders for spatial images (top) and optimal flows (bottom).

A. CNN Stream Encoder

An input video clip with two streams of spatial images ($k = 1$) and optical flows ($k = 2$), $\{\mathbf{o}_1^{(k)}, \dots, \mathbf{o}_T^{(k)}\}_{k=1}^2$ is encoded via two separate multi-layer CNNs with parameters $\{\boldsymbol{\theta}_c^{(k)}\}_{k=1}^2$, as shown in Figure 2, to calculate two streams of features $\{\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_T^{(k)}\}_{k=1}^2$, such that

$$\mathbf{x}_t^{(k)} = \text{CNN-encoder}(\mathbf{o}_t^{(k)}). \tag{1}$$

There are t time steps in a steam. The dimension of $\mathbf{x}_t^{(k)}$ is typically smaller than that of $\mathbf{o}_t^{(k)}$. Extracting two complementary streams of features is beneficial to achieve a desirable performance for image recognition. Since the configuration of CNNs for finding the embedded features $\{\mathbf{x}_t^{(k)}\}$ at the front end also plays a crucial role for the final performance, the parameters $\{\boldsymbol{\theta}_c^{(k)}\}_{k=1}^2$ were jointly trained with the subsequent RNNs for the best performance.

B. RNN Fusion Decoder

For video classification, the conventional method [5] was developed by treating the encoded features $\{\mathbf{x}_t^{(k)}\}$ as the observation vectors and then decoding two streams of feature vectors by using two separate RNNs or LSTMs so as to find the sequences of hidden codes $\{\mathbf{h}_1^{(k)}, \dots, \mathbf{h}_T^{(k)}\}_{k=1}^2$ for spatial images and optical flows. In [5], the fusion was considered only at the last time step T by transforming the last two hidden codes $\{\mathbf{h}_T^{(k)}\}_{k=1}^2$ as one, *i.e.,* considering the concatenated vector

$$\mathbf{h}_T = \left[(\mathbf{h}_T^{(1)})^\top (\mathbf{h}_T^{(2)})^\top \right]^\top \tag{2}$$

to derive the output prediction vectors

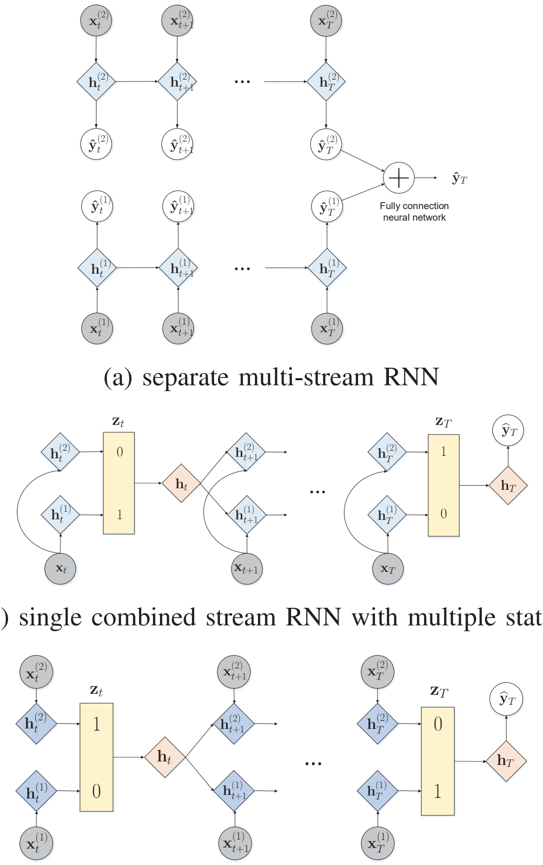
$$\widehat{\mathbf{y}}_T = \text{softmax}\left(\sigma(\mathbf{W}_y \mathbf{h}_T + \mathbf{b}_y)\right) \tag{3}$$

for the final class posteriors based on a fully connected layer $\{\mathbf{W}_y, \mathbf{b}_y\}$ as showed in Figure 3(a) and thus the final classification result was obtained. We also denote this solution as the *separate multi-stream RNN*, which disregarded the dependencies between two synchronous and complementary streams at each individual frames.

To strengthen the weakness presented in [5], we propose two approaches to combine two encoded streams $\{\mathbf{x}_t^{(1)}\}$ and $\{\mathbf{x}_t^{(2)}\}$ at each time step for video classification. The first approach is called the *single combined stream RNN with multiple states* as depicted in Figure 3(b). Using this approach, two encoded vectors are concatenated at each time frame to form a single augmented vector $\mathbf{x}_t = [(\mathbf{x}_t^{(1)})^\top (\mathbf{x}_t^{(2)})^\top]^\top$. To explore rich statistics in hybrid latent space for a sequence of augmented vectors, we allocate two latent codes $\{\mathbf{h}_t^{(1)}, \mathbf{h}_t^{(2)}\}_{t=1}^T$ associated with two discrete states $\mathbf{z}_t = \{z_t^{(k)}\}$ in a form of

$$\mathbf{h}_t = \mathbf{z}_t \odot \left(\mathbf{h}_t^{(1)}, \mathbf{h}_t^{(2)}\right) = \sum_{k=1}^{2} z_t^{(k)} \mathbf{h}_t^{(k)}, \quad \forall t \in [0, T] \tag{4}$$

where $z_t^{(k)} \in \{0, 1\}$ and $z_t^{(1)} + z_t^{(2)} = 1$ for all $t \in [0, T]$. Now Eq. (4) clearly indicates the stream fusion takes place over all temporal moments. It is obvious that Eq. (2) of [5] is only a special case of Eq. (4) where previous *stream selections* $\mathbf{z}_t \equiv 0$ with $t < T$ are not considered. In addition, it is also novelty to introduce a stochastic indicator $\mathbf{z}_t$, where the details shall be provided later in Eq. (6). In the end, the fused hidden state $\mathbf{h}_T$ is computed to provide classification prediction by $\widehat{\mathbf{y}}_T$ which represents the final class posteriors of a fully connected layer. After obtaining the prediction vector $\widehat{\mathbf{y}}_T$ with the softmax activation, the cross entropy error function between posterior output $\widehat{\mathbf{y}}_T$ and true label $\mathbf{y}_T$ over all video clips is minimized for model training. This method is comparable of running the Markov RNN [15] with Markov transition between two Markov states $\{\mathbf{h}_t^{(1)}, \mathbf{h}_t^{(2)}\}$ at each time t using $\mathbf{z}_t$.



(a) separate multi-stream RNN

(b) single combined stream RNN with multiple states

(c) multi-stream RNN with stochastic fusion

Fig. 3. Architectures for different RNN decoders.

The second proposed approach, called a *multi-stream RNN with stochastic fusion* illustrated in Figure 3(c), allows two separate encoded streams $\{\mathbf{x}_t^{(k)}\}_{k=1}^2$ to be decoded by two separate hidden codes $\{\mathbf{h}_t^{(k)}\}_{k=1}^2$ respectively via RNNs such that

$$\mathbf{h}_t^{(k)} = \sigma\left(\mathbf{W}_h^{(k)} \mathbf{x}_t^{(k)} + \mathbf{U}_h^{(k)} \mathbf{h}_{t-1}^{(k)} + \mathbf{b}_h^{(k)}\right). \tag{5}$$

Similarly, the posterior output of this approach $\widehat{\mathbf{y}}_T$ is then calculated at the final frame T. Importantly, a stochastic indicator variable $\mathbf{z}_t$ is drawn to attend or fuse the hidden codes from multiple streams based on Eq. 4 . The fusion is automatically aggregated at each time t by selecting the feature vector either from spatial image $\mathbf{x}_t^{(1)}$ or optical flow $\mathbf{x}_t^{(2)}$. Such a selection can also be regarded as a kind of masking. The classification output depends on the class posterior $\widehat{\mathbf{y}}_T$ calculated by integrating the features of spatial images and optimal flows at different frames. The fusion mechanism using $\mathbf{z}_t$ aims to integrate different streams and select one stream at each individual time t. The sequence $\{\mathbf{z}_t\}_{t=1}^T$ is drawn based on a stochastic optimization.

Finally, we remark that the *key difference* between the first and second approach is that in the former two hidden states $\{\mathbf{h}_t^{(1)}, \mathbf{h}_t^{(2)}\}$ were derived by *exactly the same input* information $\mathbf{x}_t = [(\mathbf{x}_t^{(1)})^\top (\mathbf{x}_t^{(2)})^\top]^\top$ of two streams combined, while in the latter two hidden codes $\mathbf{h}_t^{(1)}$ and $\mathbf{h}_t^{(2)}$ were fed with *different encoded streams* $\mathbf{x}_t^{(1)}$ and $\mathbf{x}_t^{(2)}$, respectively. Thus the first approach forces the two hidden states see the same thing, but require them to interpret differently out of the same input. The second approach has two hidden states to see different things with different contents and variations.

C. Stochastic Fusion and Optimization

This study presents a stochastic fusion and optimization method where the joint log likelihood of training clips is maximized. In stochastic optimization, the random fusion variable $\mathbf{z}_t$ is characterized by a categorical distribution. Given the data collection consisting of video clips $\mathbf{o}_{1:T} = \{\mathbf{o}_t\}_{t=1}^T$ and true classes $\mathbf{y}_T$, the parameters of CNN stream encoder $\boldsymbol{\theta}_c = \{\boldsymbol{\theta}_c^{(k)}\}$ and RNN fusion decoder $\boldsymbol{\theta}_r = \{\mathbf{W}_h^{(k)}, \mathbf{U}_h^{(k)}, \mathbf{b}_h^{(k)}, \mathbf{W}_y, \mathbf{b}_y, \mathbf{W}_s\}$ are estimated by maximizing the expectation of conditional log likelihood

$$\mathbb{E}_{p_{\boldsymbol{\phi}}(\mathbf{z}_{1:T}|\mathbf{o}_{1:T})}[\log p_{\boldsymbol{\theta}}(\mathbf{y}_T|\mathbf{o}_{1:T}, \mathbf{z}_{1:T})] + \sum_{t=1}^{T} \mathbb{H}[p_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{o}_t)] \tag{6}$$

or equivalently minimizing the expectation of cross entropy error function, where $\boldsymbol{\theta} = \{\boldsymbol{\theta}_c, \boldsymbol{\theta}_r\}$. In Eq. (6), the entropy function, denoted by $\mathbb{H}$, of transition probability is imposed as a regularization term in the maximization process to encourage the exploration in learning process. The expectation with respect to the stochastic fusion mask $\mathbf{z}_t$ is optimized. Following [17], [18], [15], the Gumbel-softmax is introduced to tackle the non-differentiable expectation function in inference procedure due to the sampling of discrete variable $\mathbf{z}_t$. The Gumbel-max trick with relaxation is employed to approximate the categorical distribution for discrete vector $p(\mathbf{z}_t)$ based on the reparameterization

$$\mathbf{z}_t = \text{onehot}\left(\arg\max_i \{(\log \pi_{tk} + g_{tk})/\tau\}\right) \tag{7}$$

where $\pi_{tk} \triangleq p(z_t^{(k)} = 1)$, $\{g_{tk}\}_{k=1}^2$ are i.i.d Gumbel samples $g_{tk} \sim \text{Gumbel}(0, 1)$ with zero-mean and unit-variance, and τ is the temperature for relaxation. We have $p_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{o}_t) \sim \text{Categorical}(\boldsymbol{\pi}_t)$ where a logit encoder is applied to encode the categorical parameter

$$\log(\pi_{tk}) = (\mathbf{v}^{(k)})^\top \sigma(\mathbf{W}[\mathbf{h}_{t-1}\ \mathbf{x}_t] + \mathbf{b}). \tag{8}$$

A stochastic backpropagation is fulfilled by jointly estimating CNN encoder $\boldsymbol{\theta}_c$, RNN decoder $\boldsymbol{\theta}_r$ and combiner $\boldsymbol{\phi} = \{\mathbf{v}^{(k)}, \mathbf{W}, \mathbf{b}\}$ by maximizing a differentiable objective via the Monte Carlo method [19], [20], [21].

IV. Experiments

A. Experimental Setup

UCF101 is a popular dataset for evaluation of human action recognition, which consisted of 13,320 video clips with diverse forms of camera motion and illustration from 101 action classes [10]. Each video clip has a frame rate of 25 (frames per second) with various lengths. The resolution of 320 × 240 pixels was recorded. For simplicity, we only used the first 28 frames of each video as the inputs, which is the least consensus of all videos. The CNN encoder is built up by a pretrained ResNet [22] from ImageNet with final layer retrained for the encoding purpose and fixed across all stream decoders. LSTMs [3] were used for multi-stream RNN decoder. Neural network parameters $\{\boldsymbol{\theta}_c, \boldsymbol{\theta}_r, \boldsymbol{\phi}\}$ were jointly trained with Adam optimizer [23]. We compared three methods consisting of the separate multi-stream RNN, the single combined stream RNN with multiple states, and the multi-stream RNN with stochastic fusion.

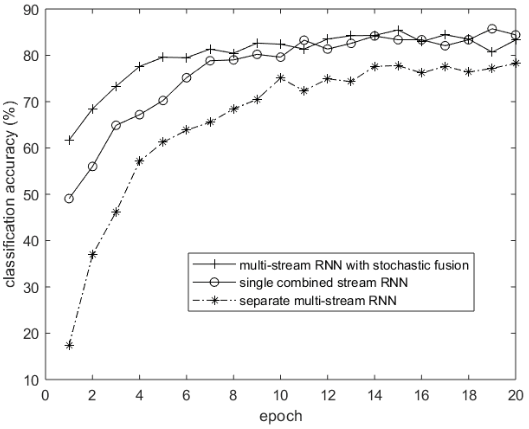


Fig. 4. Test accuracy versus learning epoch by using different methods.

B. Experimental Results

Figure 4 shows the comparison of learning curves in terms of test accuracy for the first 20 epochs. The single combined stream RNN and multi-stream RNN converge much better than the separate multi-stream RNN. Table I shows the classification performance and the number of parameters used in different methods. Essentially, the training batch size was fixed at 40 and initial learning rate was set at 10^{-3}. The multi-stream fusion RNN performs better than single stream RNN and separate multi-stream RNN. Although the multi-stream RNN achieves comparable accuracy with single combined stream RNN, the size of parameters using multi-stream RNN has been greatly reduced. The overhead of the number of parameters is moderate by using the proposed multi-stream

RNN with stochastic fusion relative to the baseline system based on separate multi-stream RNN.

	# of params	accuracy
single stream RNN	0.9M	74.1%
separate multi-stream RNN	1.9M	78.2%
single combined stream RNN	5.5M	85.5%
multi-stream RNN & stochastic fusion	2.4M	85.7%

TABLE I
PARAMETER SIZE AND CLASSIFICATION ACCURACY BY USING DIFFERENT METHODS.

Table II shows the percentages of samples in different streams after the training procedure of multi-stream RNN has converged. Stochastic attention over time steps is displayed to manifest the focus of certain stream at different frames. We also had some interesting observations in certain actions. For example, Figure 5 and 6 illustrate several examples of video clips and the corresponding categorical distributions, $p_{\phi}(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{o}_t)$. In *bench press* or *soccer juggling*, since the barbell and soccer move up and down more frequently, the sampled categorical probability of optical flow in the y-axis is higher than that in other actions. We also observed similar case in jumping-related action such as *jumping jack* or *jumping rope*. On the other hand, on gymnastics such as *parallel bars* and *still rings*, the sampled categorical probabilities of optical flow x and y are comparable. As we can see, our fusion model combining multiple streams is capable of understanding the utility and meaning of different streams.

stream	spatial	optical x	optical y
percentage	71%	9%	20%

TABLE II
PERCENTAGES OF SAMPLES IN DIFFERENT STEAMS AMONG ALL TRAINING VIDEO CLIPS.

V. CONCLUSIONS

In this work, we have presented a new framework on multi-stream RNN for video classification. We explored the stochastic fusion mechanism on the hidden states of multi-stream data such that it provided a meaningful and interpretable way for fusing information. We showed the effectiveness of the proposed model on UCF101 dataset in terms of stream visualization, stream occupation, learning curve and classification accuracy. By taking the advantage of complementary information, our method outperformed the previous works with limited size of parameters. We also found interesting interpretation about how streams were utilized by peeking into the fusion weights. We deem this as a valuable direction to visualize how the information from different sources is integrated by the fusion mechanism. This method will be generalized for other type of technical data.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

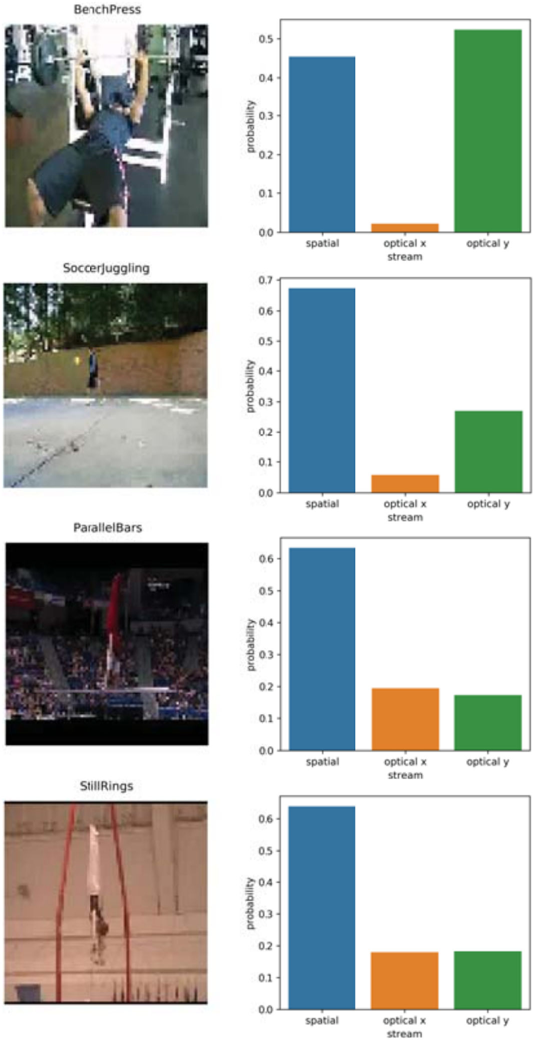


Fig. 5. Examples of video clips and their categorical probabilities.

[2] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[5] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[6] J.-T. Chien and C.-W. Ting, "Acoustic factor analysis for streamed hidden Markov modeling," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 7, pp. 1279–1291, 2009.

[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. of Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[8] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. of Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933–1941.

[9] M. Bouaziz, M. Morchid, R. Dufour, G. Linarès, and R. De Mori, "Parallel long short-term memory for multi-stream classification," in *Proc. of IEEE Spoken Language Technology Workshop*, 2016, pp. 218–223.

[10] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, X. Xue, and J. Wang, "Fusing multi-stream deep networks for video classification," *arXiv preprint arXiv:1509.06086*, 2015.

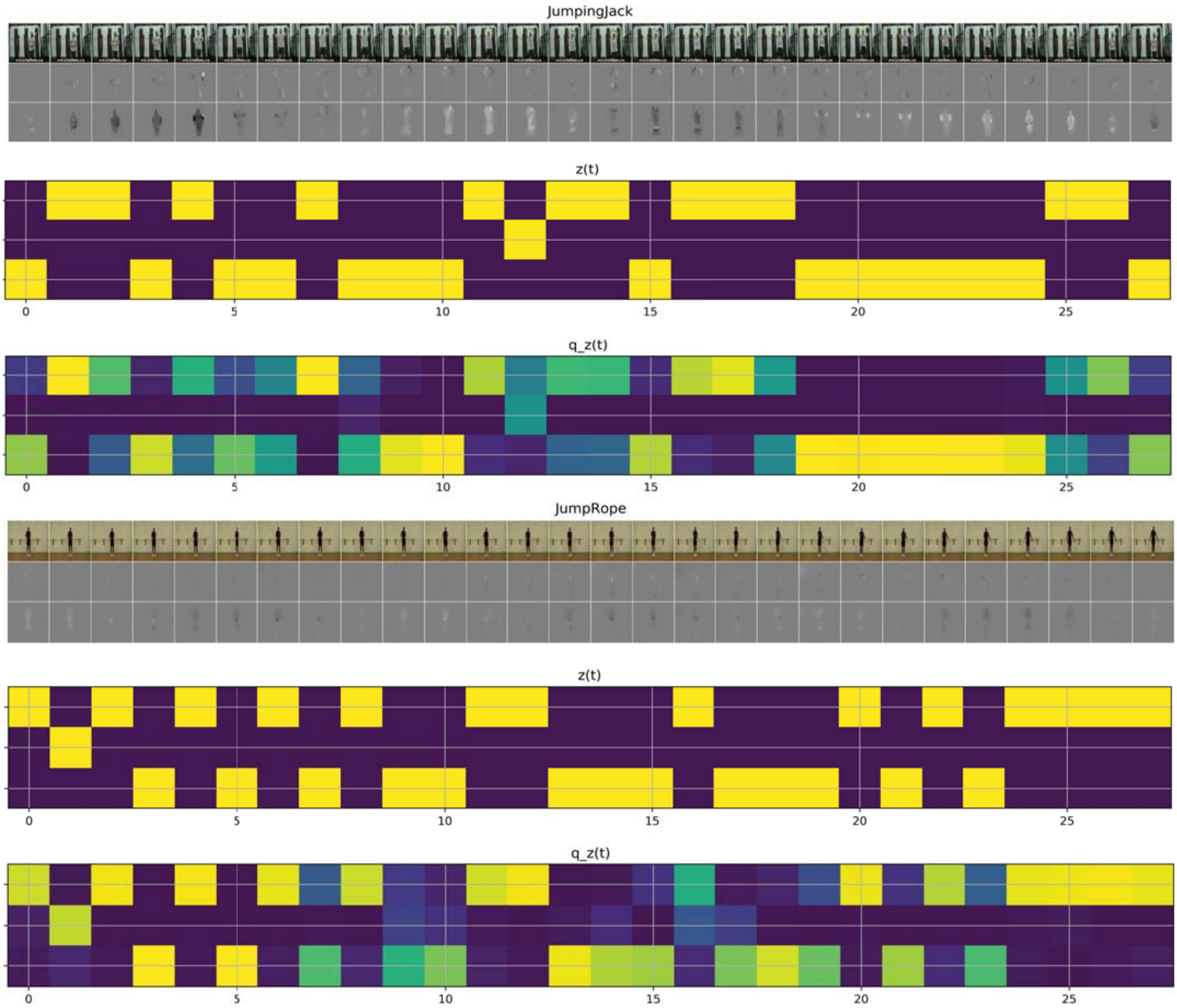


Fig. 6. Examples of two video clips. Row 1 displays the spatial images and rows 2 and 3 display the optical flows x and y. Mostly likely streams are shown by yellow. Yellow means one while purple means zero. Rows 4 to 6 show the categorical probabilities of $\mathbf{z}_t$. Horizontal axis shows 28 time frames.

[11] M. Bouaziz, M. Morchid, R. Dufour, and G. Linares, "Improving multi-stream classification by mapping sequence-embedding in a high dimensional space," in *Proc. of IEEE Spoken Language Technology Workshop*, 2016, pp. 224–231.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[13] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. of International Conference on Machine Learning*, 2015, pp. 2048–2057.

[14] X. Long, C. Gan, G. de Melo, X. Liu, Y. Li, F. Li, and S. Wen, "Multimodal keyless attention fusion for video classification," 2018.

[15] C.-Y. Kuo and J.-T. Chien, "Markov recurrent neural networks," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, 2018, pp. 1–6.

[16] J.-T. Chien and C.-Y. Kuo, "Stochastic markov recurrent neural network for source separation," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 8072–8076.

[17] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.

[18] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[19] J.-T. Chien, "Deep Bayesian mining, learning and understanding," in *Proc. of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 3197–3198.

[20] J.-T. Chien, "Hierarchical theme and topic modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 3, pp. 565–578, 2016.

[21] Jen-Tzung Chien and Chao-Hsi Lee, "Deep unfolding for topic models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 318–331, 2018.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.