Action Recognition using Visual Attention

Shikhar Sharma **Ryan Kiros** **Ruslan Salakhutdinov**
Department of Computer Science, University of Toronto
{shikhar,rkiros,rsalakhu}@cs.toronto.edu

Abstract

We propose a soft attention based model for the task of action recognition in videos. We use multi-layered Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) units which are deep both spatially and temporally. Our model learns to focus selectively on parts of the video frames and classifies videos after taking a few glimpses. The model essentially learns which parts in the frames are relevant for the task at hand and attaches higher importance to them. We evaluate the model on UCF-11 (YouTube Action), HMDB-51 and Hollywood2 datasets and analyze how the model focuses its attention depending on the scene and the action being performed.

1 Introduction

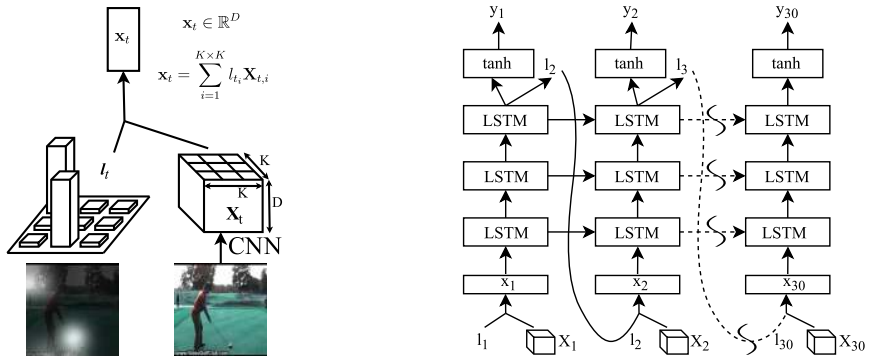
It has been noted in visual cognition literature that humans do not focus their attention on an entire scene at once. Instead, they focus sequentially on different parts of the scene to extract relevant information. With the recent surge of interest in deep neural networks, attention based models have been shown to achieve promising results on several challenging tasks. Attention models can be classified into soft attention and hard attention models. Soft attention models are deterministic and can be trained using backpropagation, whereas hard attention models are stochastic and can be trained by the REINFORCE algorithm, or by maximizing a variational lower bound or using importance sampling [1]. Attention based models can potentially infer the action happening in videos by focusing only on the relevant places in each frame.

Convolutional Neural Networks (CNNs) have been highly successful in image classification and object recognition tasks [8]. Classifying videos instead of images adds a temporal dimension to the problem of image classification. Learning temporal dynamics is a difficult problem and earlier approaches have used optical flow, HOG and hand-crafted features to generate descriptors with both appearance and dynamics information encoded. LSTMs [6] have been recently shown to perform well in the domain of machine translation [12] and image description [14]. More recently, [15] has proposed to use 3-D CNN features and an LSTM decoder in an encoder-decoder framework to generate video descriptions. Their model incorporates attention on a video level by defining a probability distribution over frames used to generate individual words.

In general, it is rather difficult to interpret internal representations learned by deep neural networks. Attention models add a dimension of interpretability by capturing where the model is focusing its attention when performing a particular task. A recent work of [14] used both soft attention and hard attention mechanisms to generate image descriptions. Their model actually looks at the respective objects when generating their description. Our work directly builds upon this work. However, while [14] primarily worked on caption generation in static images, in this paper, we focus on using a soft attention mechanism for activity recognition in videos. We further demonstrate that our model tends to recognize important elements in video frames based on the activities it detects.

2 The Model and the Attention Mechanism

We extract the last convolutional layer obtained by pushing the video frames through GoogLeNet model [13] trained on the ImageNet dataset [4]. This last convolutional layer has D convolutional



(a) The soft attention mechanism

(b) Our recurrent model

Figure 1: (1a) The CNN takes the video frame as its input and produces a feature cube. The model computes the current input $\mathbf{x}_t$ as an average of the feature slices weighted by the location softmax l_t (1b) At each time-step t, our recurrent network takes a feature slice $\mathbf{x}_t$, generated as in (1a), as the input. It then propagates $\mathbf{x}_t$ through three layers of LSTMs and predicts the next location probabilities l_{t+1} and the class label y_t.

maps and is a feature cube of shape $K \times K \times D$ ($7 \times 7 \times 1024$). Thus, at each time-step t, we extract K^2 D-dimensional vectors. We refer to these vectors as feature slices in a feature cube:

$$\mathbf{X}_t = [\mathbf{X}_{t,1}, \ldots, \mathbf{X}_{t,K^2}], \qquad \mathbf{X}_{t,i} \in \mathbb{R}^D.$$

Each of these K^2 vertical feature slices maps to different overlapping regions in the input space and our model chooses to focus its attention on these K^2 regions. We use the LSTM implementation discussed in [16] and [14]. At each time-step t, our model predicts $\mathbf{l}_{t+1}$, a softmax over $K \times K$ locations, and $\mathbf{y}_t$, a softmax over the label classes with an additional hidden layer with $tanh$ activations (see Fig. 1b). The location softmax is defined as follows:

$$l_{t_i} = p(\mathbf{L}_t = i|\mathbf{h}_{t-1}) = \frac{\exp(W_i^\top \mathbf{h}_{t-1})}{\sum_{j=1}^{K\times K}\exp(W_j^\top \mathbf{h}_{t-1})} \qquad i \in 1 \ldots K^2, \tag{1}$$

where W_i are the weights mapping to the i^{th} element of the location softmax, $\mathbf{h}_{t-1}$ is the hidden state at time-step $t-1$ and $\mathbf{L}_t$ is a random variable which can take 1-of-K^2 values. This softmax can be thought of as the probability with which our model believes the corresponding region in the input frame is important. After calculating these probabilities, the soft attention mechanism [2] computes the expected value of the input at the next time-step $\mathbf{x}_t$ by taking expectation over the feature slices at different regions (see Fig. 1a):

$$\mathbf{x}_t = \mathbb{E}_{p(\mathbf{L}_t|\mathbf{h}_{t-1})}[\mathbf{X}_t] = \sum_{i=1}^{K\times K} l_{t_i}\mathbf{X}_{t,i}, \tag{2}$$

where $\mathbf{X}_t$ is the feature cube and $\mathbf{X}_{t,i}$ is the i^{th} slice of the feature cube at time-step t. In our experiments, we use multi-layered deep LSTMs, as shown in Fig. 1b.

We use cross-entropy loss coupled with the doubly stochastic penalty introduced in [14]. We impose an additional constraint over the location softmax, so that $\sum_{t=1}^{T} l_{t_i} \approx 1$. This is the attention regularization which forces the model to look at each region of the frame at some point in time.

3 Experiments

We have used UCF-11, HMDB-51 and Hollywood2 datasets in our experiments. Each video in UCF-11 and HMDB-51 has only one action associated with it while some videos in Hollywood2 have multiple actions associated with them. All the videos in the datasets were resized to 224×224 resolution and fed to a GoogLeNet model trained on the ImageNet dataset. The last convolutional layer of size $7 \times 7 \times 1024$ was used as an input to our model.

Our implementation is based in Theano [3] which also handles the gradient computation. For both training and testing our model takes 30 frames at a time sampled at fixed fps rates. We compute class predictions for each time step and then average those predictions over 30 frames. To obtain a prediction for the entire video clip, we average the predictions from all 30 frame blocks in the video. Among baselines, the softmax regression model uses the complete $7 \times 7 \times 1024$ feature cube as its input to predict the label at each time-step t, while all other models use only a 1024-dimensional feature slice as their input. The average pooled and max pooled LSTM models use the

Table 1: Performance on UCF-11 (acc %), HMDB-51 (acc %) and Hollywood2 (mAP %)

Model	UCF-11	HMDB-51	Hollywood2
Softmax Regression (full CNN feature cube)	82.37	33.46	34.62
Avg pooled LSTM (@ 30 fps)	82.56	40.52	43.19
Max pooled LSTM (@ 30 fps)	81.60	37.58	43.22
Soft attention model (@ 30 fps)	84.96	41.31	43.91

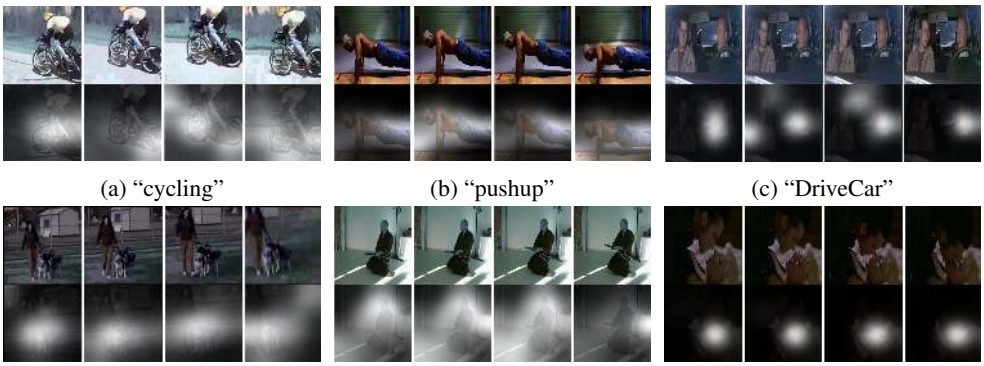
Table 2: Comparison of performance on HMDB-51 and Hollywood2 with state-of-the-art models

Model		HMDB-51 (acc %)	Hollywood2 (mAP %)
Spatial stream ConvNet	[9]	40.5	-
Soft attention model	(Our model)	41.3	43.9
Composite LSTM Model	[10]	44.0	-
DL-SFA	[11]	-	48.1
VideoDarwin	[5]	63.7	73.7
Objects+Traditional+Stacked Fisher Vectors	[7]	71.3	66.4

same architecture as our model except that they do not have any attention mechanism and thus do not produce a location softmax. The inputs at each time-step for these models are obtained by doing average or max pooling over the $7 \times 7 \times 1024$ cube to get 1024 dimensional slices, whereas our soft attention model dynamically weights the slices by the location softmax (see Eq. 2).

3.1 Quantitative analysis

Table 1 reports accuracies on UCF-11 and HMDB-51 datasets and mean average precision (mAP) on Hollywood2. The results from Table 1 demonstrate that our attention model performs better than both average and max pooled LSTMs, and the softmax regression baseline. In Table 2, we compare the performance of our model with other state-of-the-art action recognition models. We have divided the table into three sections. Models in the first section use only RGB data while the model in the second section use both RGB and optical flow data. The model in the third section uses both RGB, optical flow, as well as object responses of the videos on some ImageNet categories. Our model performs competitively against other deep learning models in its category (models using RGB features only), while providing some insight into where the neural network is looking.



(a) "cycling" (b) "pushup" (c) "DriveCar"

(d) "walking with a dog" (e) "draw_sword" (f) "Kiss"

Figure 2: Correctly classified video frames showing attention over time: The white regions are where the model is looking and the brightness indicates the strength of focus. The model learns to look at relevant parts.

(a) Incorrectly classified as "somersault" (b) Incorrectly classified as "diving" (c) Incorrectly classified as "volleyball"

Figure 3: Incorrectly classified video frames showing attention over time: The white regions are where the model is looking. Different glimpses can result in different predictions. Best viewed in color.

3.2 Qualitative analysis

Fig. 2 and Fig. 3 show some test examples of where our model attends to over time. It is quite interesting to see that we can better understand the success and failure cases of this deep attention model by visualizing where it attends to.[1] We can see that to classify the corresponding activities correctly, the model focuses on parts of the cycle in Fig. 2a, the person doing push-ups in Fig. 2b, and the steering wheel, the rear-view mirror, and the men in Fig. 2c. Fig. 2d, shows an example of a short clip belonging to "walking with a dog" activity and our model attends to the dogs and the person. Similarly, in Fig. 2e, it focuses on the swordsman drawing out his sword. In Fig. 2f, the model correctly anticipates that a "Kiss" is going to take place and attempts to focus on the region between the man and the woman.

The model, however, is not perfect. Among failure cases, Fig. 3b shows an example where the model mostly attends to the background like the light blue floor of the court. The model incorrectly classifies the example as "diving" instead of "soccer juggling". Using a different glimpse, as shown in Fig. 3c, the model classifies the same example as "volleyball spiking". In Fig. 3a we can see that the model classifies the example of "kick_ball" incorrectly as "somersault" despite attending to the location where the action is happening.

4 Conclusion

In this paper we developed recurrent soft attention based models for action recognition and tried to interpret where they focus their attention. We presented both quantitative and qualitative analysis of the results we obtained. Our proposed model tends to recognize important elements in video frames based on the action that is being performed. We also showed that our model performs better than baselines which do not use any attention mechanism. These models can also be extended to the multi-resolution setting, in which the attention mechanism could also choose to focus on the earlier convolutional layers in order to attend to the lower-level features in the video frames.

References

[1] J. Ba, R. Grosse, R. Salakhutdinov, and B. Frey. Learning wake-sleep recurrent attention models. In *Advances in Neural Information Processing Systems 28*, 2015.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

[3] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio. Theano: new features and speed improvements. *CoRR*, abs/1211.5590, 2012.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[5] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015.

[6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8), 1997.

[7] M. Jain, J. C. v. Gemert, and C. G. M. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *CVPR*, June 2015.

[8] S. Ren, K. He, R. B. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. *CoRR*, abs/1504.06066, 2015.

[9] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *CoRR*, abs/1406.2199, 2014.

[10] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using LSTMs. *ICML*, 2015.

[11] L. Sun, K. Jia, T.-H. Chan, Y. Fang, G. Wang, and S. Yan. DL-SFA: deeply-learned slow feature analysis for action recognition. In *CVPR*, 2014.

[12] I. Sutskever, O. Vinyals, and Q. V. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. 2014.

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

[14] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *ICML*, 2015.

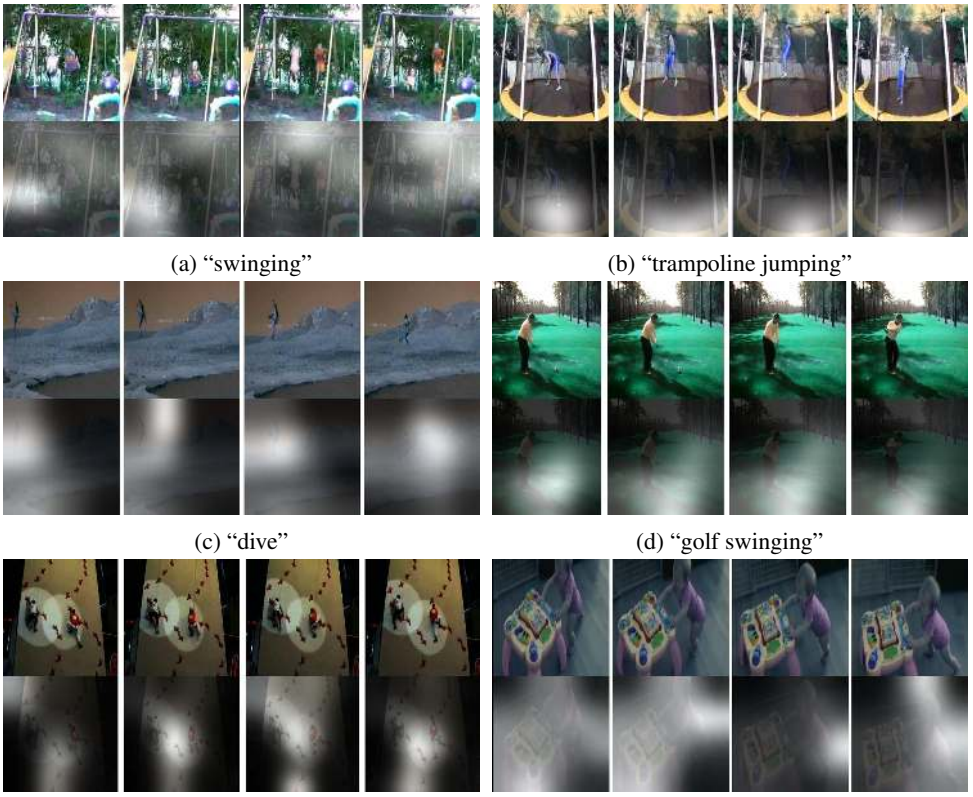
[15] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. *CoRR*, abs/1502.08029, 2015.

[16] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *CoRR*, abs/1409.2329, 2014.

[1] More examples of our model's attention are available in Appendix A and at `http://www.cs.toronto.edu/~shikhar/projects/action-recognition-attention`.

A Additional examples

We present some more correctly classified examples in Fig. 4 and incorrectly classified examples in Fig. 5.



(a) "swinging" (b) "trampoline jumping"

(c) "dive" (d) "golf swinging"

(e) "climb" (f) "push"

Figure 4: Correctly classified video frames showing attention over time: The white regions are where the model is looking and the brightness indicates the strength of focus. The model learns to look at relevant parts.

(a) "flic_flac" misclassified as "hit" (b) "laugh" misclassified as "smile"

Figure 5: Incorrectly classified video frames showing attention over time: The white regions are where the model is looking and the brightness indicates the strength of focus.

B The Model and the Attention Mechanism

B.1 The LSTM

The LSTM implementation we use in this paper is as follows:

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} M \begin{pmatrix} \mathbf{h}_{t-1}, \\ \mathbf{x}_t \end{pmatrix}, \tag{3}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \tag{4}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \tag{5}$$

where $\mathbf{i}_t$ is the input gate, $\mathbf{f}_t$ is the forget gate, $\mathbf{o}_t$ is the output gate, and $\mathbf{g}_t$ is calculated as shown in Eq. 3. $\mathbf{c}_t$ is the cell state, $\mathbf{h}_t$ is the hidden state, and $\mathbf{x}_t$ represents the input to the LSTM at time-step t. $M : \mathbb{R}^a \rightarrow \mathbb{R}^b$ is an affine transformation consisting of trainable parameters with $a = d + D$ and $b = 4d$, where d is the dimensionality of all of $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$, $\mathbf{g}_t$, $\mathbf{c}_t$, and $\mathbf{h}_t$.

We initialize the cell state and the hidden state of the LSTM as:

$$\mathbf{c}_0 = f_{\text{init,c}}\left(\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{K^2}\sum_{i=1}^{K\times K}\mathbf{X}_{t,i}\right)\right) \quad \text{and} \quad \mathbf{h}_0 = f_{\text{init,h}}\left(\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{K^2}\sum_{i=1}^{K\times K}\mathbf{X}_{t,i}\right)\right), \quad (6)$$

where $f_{\text{init,c}}$ and $f_{\text{init,h}}$ are two multilayer perceptrons and T is the number of time-steps in the model. These values are used to calculate the first location softmax $\mathbf{l}_1$ which determines the initial input $\mathbf{x}_1$.

B.2 Loss Function and the Attention Penalty

The loss function is defined as follows:

$$L = -\sum_{t=1}^{T}\sum_{i=1}^{C} y_{t,i}\log\hat{y}_{t,i} + \lambda\sum_{i=1}^{K\times K}(1-\sum_{t=1}^{T} l_{t_i})^2 + \gamma\sum_{i}\sum_{j}\theta_{i,j}^2, \quad (7)$$

where y_t is the one hot label vector, $\hat{y}_t$ is the vector of class probabilities at time-step t, T is the total number of time-steps, C is the number of output classes, λ is the attention penalty coefficient, γ is the weight decay coefficient, and θ represents all the model parameters.

C Analysis

C.1 Experiments with the Attention Penalty and Framerate

We experimented with the doubly stochastic penalty term λ (see Eq. 7) as well. Fig. 6 shows that with no attention regularization term, $\lambda = 0$, the model tends to select a few specific locations and stay fixed on them while setting $\lambda = 1$ encourages the model further explore different gaze locations.

It is also interesting to observe that in some cases, the model is able to attend to important objects in the video frames and attempts to track them to some extent. In Fig. 7b, the video is sampled at 30fps and the model stays focused on the golf ball, club, and the human. However, when we change the sampling rate to 6fps, as shown in Fig. 7a, we find that the video frames change quickly. The model now remains focused on the ball before it disappears. After the person hits the ball, we see that the model tries to look at other places.

(a) $\lambda = 0$ (b) $\lambda = 1$

Figure 6: Variation in the model's attention depending on the value of attention penalty λ. The white regions are where the model is looking and the brightness indicates the strength of focus.

(a) "golf swinging" (@ 6fps, $\lambda = 1$) (b) "golf swinging" (@ 30 fps, $\lambda = 1$)

Figure 7: The model's focus of attention visualized over four equally spaced time-steps at different fps rates. (a) plays faster and when the ball is hit and the club disappears, the model searches around to find them. (b) plays slower and the model stays focused on the ball and the club.