Proceedings of the
7th Graz Brain-Computer Interface Conference 2017 DOI: 10.3217/978-3-85125-533-1-87

USING RECURRENT NEURAL NETWORKS FOR P300-BASED BRAIN-COMPUTER INTERFACES

O. Tal, D. Friedman

The Advanced Reality Lab, The Interdisciplinary Center, Herzliya, Israel

Contact: doronf@idc.ac.il

ABSTRACT: P300-based spellers are one of the main methods for electroencephalogram (EEG)-based brain-computer interface, and the detection of the target event with high accuracy is an important prerequisite. The rapid serial visual presentation (RSVP) protocol is of high interest because it can be used by patients who have lost control over their eyes. In this study we wish to explore the suitability of recurrent neural networks (RNNs) as a machine learning method for identifying the target letter in RSVP data. We systematically compare RNN with alternative methods such as linear discriminant analysis (LDA) and convolutional neural networks (CNN). Our results indicate that RNN does not have any advantages in single subject classfication. However, we show that a network combining CNN and RNN is superiour in transfer learning among subjects, and is significantly more resilient to temporal noise than other methods.

INTRODUCTION

Neural networks have recently been shown to achieve outstanding performance in several machine learning domains such as image recognition [15] and voice recognition [12]. Most of these breakthroughs have been achieved with CNNs [16], but some promising results have also been demonstrated by using RNNs for tasks such as speech and handwriting recognition [11, 10], usually when using the long short-term memory (LSTM) architecture [13]. CNNs are feed forward networks that implement receptive fields. RNNs, on the other hand, contain directed cycles and are thus able to "remember" the previous activation state of the network, which makes them especially suitable for learning sequences.

There have been some studies on using "deep neural networks" for P300 classification [5, 19]. The results reported, despite some success, do not show the same dramatic progress achieved by 'deep learning' methods as compared to the previous state of the art; while in areas such as image or voice recognition 'deep' neural networks have resulted in classification accuracy exceeding other methods by far, this has not yet been the case with EEG in general and P300 detection specifically. The small number of samples typically available in neuroscience (or BCI) is most likely one of the main reasons. In addition, the high dimensionality of the EEG signal, the low signal to noise (SNR) and the existence of outliers in the data, pose other difficulties when trying to use neural networks for BCI tasks (see [18]). The main question in this research is whether the RNN model, and particularly LSTM, can enhance the accuracy of P300-based BCI systems and if so, under what conditions.

BACKGROUND

P300-based BCI systems can recognize a taregt stimulus out of a set of stimuli, typically letters and numbers, by examining the subject's EEG data. The first system that used the P300 effect was presented by [8] and since then different versions of P300 based BCI systems were suggested. One example of such a paradigm is the P300 rapid serial visual presentation (RSVP) speller. In this paradigm letters are presented one after the other in a random order, and the subject is asked to pay attention only to one of the letters, reffered to as the *target* (e.g., by counting them silently).

There are a lot of methods for identifying the target letter for a BCI task. Blankertz et al. [4] suggest to select the time interval with maximal separation between the target and non target samples, average their electro-potential value and use shrinkage LDA to classify these features. Using this method has a drawback due to the low complexity of LDA model [6]. The winner of the BCI competition III: dataset II used an ensemble of support vector machines (SVM) [21], and other methods include hidden Markov model, k-nearest neighbours, and more [6].

More recently, given the success of 'deep' neural networks [15], there have been several attempts to apply 'deep learning' for BCI related tasks. Cecotti and Graser [5] were the first to use CNNs for a P300 speller. In their work, they train an ensemble of CNN-based P300 classifiers to identify the existence of P300. Manor and Geva [19] used CNN for the RSVP P300 classification task and suggested a new spatio-temporal regularization method, which have shown improvement in the performance.

Unlike feed forward network models such as CNN and multi-layer perceptron (MLP), the RNN architecture allows directed cycles within the network, which enable

 DOI: 10.3217/978-3-85125-533-1-87

the model to "memorize past events". LSTM [13] is a type of RNN, which includes a special node that can be described as a differentiable memory cell. The specific architecture of LSTM enables it to overcome some of the weakness of simple RNNs [3].
There are several reasons why LSTM is a good candidate for modelling the P300 pattern. First, RNN and LSTM have shown success when modeling time series for tasks such as handwriting and speech recognition [11, 10, 28]. Second, RNN is known to have the capability to approximate dynamical systems [17], which makes it a natural candidate for modelling the dynamics of EEG data. Another motivation is that RNN can be seen as a powerful form of hidden Markov models (HMM), which have been shown to classify EEG successfully [23, 20, 6]; RNNs can be seen as HMMs with an exponentially large state space and an extremely compact parametrization [24].
LSTM was already used for analysing EEG data for emotion detection [22] and a phenomena called behavioral microsleeps [7]. Bahshivan et al. [2] modeled inter-subject EEG features for identifying cognitive load by using convolutional LSTM. Their representation of the input was a "video" comprised of topographic scalp maps in three different band powers over time. One of the major differences between their work and ours is that we use the original signal without any feature extraction (such as band power), and we focus specifically on P300 speller data.

MATERIALS AND METHODS

We compared the performance of LSTM based methods with other methods on a dataset from a RSVP P300 speller study [1]. We used average prediction across 10 trials to measure the P300 speller accuracy as applied in [1].
The dataset includes 55 channels of EEG recordings from 11 subjects. Each subject is presented with 10 repetitions of 60 to 70 sets of 30 different letters and symbols. In total there are approximately 20,000 samples for each subject where 1/30 of them are supposed to contain a P300 wave. While the original experiment contains 3 different settings (interval of 116ms with/without colors and 83ms with color), we used the experiment setting of 116ms intervals with letters in different colors. For more detail, see [1].
In addition to the filters applied in [1], all models that we used share the same pre-processing stage of downsampling the input frequency from 200Hz to 25 Hz. The result is that each learning sample is a matrix of 55 channels with 25 time samples each, or $55 * 25 = 1375$ features. Each sample thus covers exactly 1 second around the target event, at times [-200,800] ms.
The models evaluated in this experiment are:

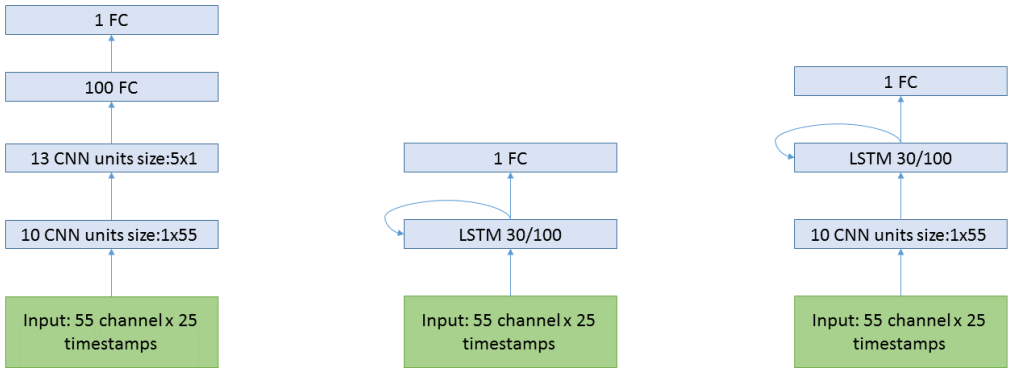
- LDA - A common method used in P300 classification for BCI [1, 4]. Here we used a simplified version; unlike [1] we use all the timestamps as features, and we are using a non-shrinkage version of LDA.
- CNN (Fig.1a) – The CNN model we use is similar to the one used in [5]. The first layer is composed of 10 spatial filters, each of size $55 * 1$ – the number of channels. The second layer contains 13 different temporal filters with size of $1 * 5$. Each one of the temporal filters processes 5 subsequent time stamps without overlapping. The third and fourth layers are simple fully connected layers followed by a single cell with a sigmoid activation function that emits a scalar.
- LSTM large/small (Fig.1b) – LSTM large/small are both composed of single LSTM layers with 100 and 30 hidden cells in each, correspondingly. Both models end with a single cell with a sigmoid activation layer that emits a scalar.
- LSTM-CNN large/small Fig.1c – The model has CNN as a first layer (the spatial domain layer) and LSTM as the second layer for the temporal domain. The first convolutional layer is the same as in the CNN model. Unlike the CNN model, the temporal layer is an LSTM layer with 100/30 hidden cells. The last layer contains a single cell with a sigmoid activation layer that emits a scalar.

In order to examine the power of each method in modelling the inter-subject and intra-subject variance we have conducted the following experiments:

1. Training and testing on each subject's data separately in order to explore intra-subject generalization.
2. Training and testing on all the different subjects data combined in order to investigate the impact of larger amounts of data.
3. Training on all subjects expect one. We conduct this experiment in order to explore the performance of a model that was trained off-line, on different subjects, and then applied to a new subject, with or without additional calibration, as a test of transfer learning.
4. Testing different models when introducing temporal noise.

A highly desired property from BCI systems is tolerance to a small degree of noise in the stimuli onset time, and this is the objective of the fourth experiment. In order to evaluate the resistance to such noise, we use a model trained on the original stimuli onset (i.e, noise level = 0ms) and evaluate its performance on different stimuli onset: noise levels of -120ms,-80ms,-40ms, +40ms, +80ms, and 120ms. We conducted this experiment using 10-fold cross validation in order to be able to get statistically significant results. This last experiment was conducted only on the CNN and LSTM-CNN models and

 DOI: 10.3217/978-3-85125-533-1-87



(a) CNN model (b) LSTM model (c) CNN-LSTM model

Figure 1: Schematic diagrams of the neural networks evaluated. FC stands for fully connected layers.

used data from all subjects (as in experiment 2 described above).

For all the experiments, the different models were trained using the RMSProp [26] optimizer: first for 30 epochs with a learning rate of 0.001 and then continued training for additional 30 epochs with a learning rate of 0.00001.

RMSProp [26] is a stochastic gradient descent (SGD) method. Unlike simple SGD, the method adapts a different learning rate for each parameter separately by applying a moving average across the magnitude of the past gradients, and then re-scaling the learning by this past gradient. We decided to use RMSProp since it is said to be robust and fast [27, 14, 25].

RESULTS

Tab. 1 summarizes the results of the different experiments; all results are based on an average of 10 consecutive trials to detect the target letter, as in [1]. The results for training and testing on the same subject indicate that LSTM is inferior (82%), and even the LSTM-CNN combined model performs less than the the simple LDA method (86 and 93% in the LSTM-CNN models and 96% using LDA) . A possible advantage for LSTM only becomes apparent with larger amounts of data – when training and testing on all the subjects together (Tab. 1). The large LSTM model performs poorly – 77%; we suspect that this is due to the large number of trainable parameters – 62501 ("over-fitting"); this is why we introduced CNN as a first layer and reduced the number of hidden LSTM cells.

Tab. 2 summarizes the results per single subject. There is a significant difference among subjects, across the different models. For example subject *fat* results in higher accuracy than *icn* regardless of the tested model. Eventually, the best network method - using training on other subjects and recalibration with a combined CNN-LSTM large model, is able to boost the results of the worse subject to 86%.

Tab. 3 is aimed at estimating learning across subjects – it provides the detailed results when training the models on all subjects except one, and then testing on that subject. In the second stage, we continue training the model on the rest $3/4$ of the **test subject's** data using a smaller learning rate (0.0001 using RMSProp) for 30 epochs – this is presented in columns *CNN and LSTM-CNN all except one fine tune* . The results indicate that the LDA accuracy is much poorer than those of the CNN and LSTM-CNN models (65% as opposed to 84%); i.e., the neural networks are superior under these conditions of inter-subject variability. When we allow calibrating the model for each subject, we achieve an average accuracy of 97% for both CNN and LSTM-CNN; there is no standard method for similarly re-calibrating an LDA algorithm, so we do not have an equivalent comparison.

Resistance to temporal noise is displayed in Tab. 4. In this test we also see that LDA accuracy drops significantly. Both CNN and the combined LSTM-CNN seems to overcome such noise; the LSTM-CNN model results in 4% or 5% when adding or removing 40ms to the original stimuli onset, and a t-test indicates that this difference is statistically significant ($p < 0.05$).

A possible explanation can be seen when looking at the two models' saliency map (Fig. 2). In order to investigate the "attention", or the sensitivity of the LSTM model, and compare it to the CNN model, we used a technique suggested by [9] and draw the absolute gradient of the neural network with respect to the input.

If $f(x_1, ..., x_n)$ is a differentiable, scalar-valued function, its gradient is the vector whose components are the n partial derivatives of f, which is a vector-valued function. In our case of $f(x|\theta)$ is the neural network with fixed weights θ and input x. The partial derivatives of $f(x|\theta)$ with respect to x can be interpreted as "how changing each value of x will change the prediction score". This gradient should not be confused with the gradient used for training, where the goal is to optimize the model parameters θ when x is fixed.

 DOI: 10.3217/978-3-85125-533-1-87

Table 1: Average accuracy across all experiments; x marks experiments that were not performed.

model	number of parameter	accuracy per subjects	accuracy all subjects	all but one	all but one after fine tuning
LDA	1375	0.96	0.79	0.65	x
LSTM large	62501	0.82	0.77	x	x
LSTM small	10351	0.86	0.9	x	x
CNN	7924	0.98	0.92	0.84	0.97
LSTM-CNN large	49041	0.93	0.9	x	x
LSTM-CNN small	5511	0.89	0.93	0.84	0.97

Table 2: Average accuracy per subject comparing all models.

subject	LDA	LSTM large	LSTM-CNN large	CNN	LSTM small	LSTM-CNN small
fat	1.00	0.98	0.98	0.98	1.00	0.95
gcb	0.91	0.82	0.88	0.92	0.74	0.75
gcc	1.00	0.84	0.92	1.00	0.92	0.97
gcd	0.97	0.80	0.90	1.00	0.76	0.93
gcf	1.00	0.92	0.94	0.95	0.97	0.95
gcg	0.94	0.74	0.96	0.96	0.80	0.87
gch	0.97	0.93	0.96	0.97	0.97	0.96
iay	0.94	0.62	0.92	0.98	0.75	0.86
icn	0.94	0.62	0.86	0.98	0.77	0.77
icr	0.93	0.97	0.98	0.98	0.98	0.98
pia	0.97	0.82	0.94	1.00	0.77	0.81
mean	0.96	0.82	0.93	0.98	0.86	0.89

In the case of P300 prediction, x is a matrix of $C \times T$ (C - number of channels, T - number of time steps) and $f(x|\theta)$ is the neural network where θ is the model's weights after training. The gradient $\nabla f(x|\theta)$ (see Eq.1) is a matrix with the same size as the input x, where the amplitude of each cell reflects its impact on the function value. Cells with high absolute value can be interpreted as the cells that have a significant influence on the prediction function.

$$\nabla f\left(x|\theta\right) = \begin{bmatrix} \frac{\partial f(x|\theta)}{\partial x(c_1,t_1)} & \dots & \frac{\partial f(x|\theta)}{\partial x(c_1,t_T)} \\ \dots & \dots & \dots \\ \frac{\partial f(x|\theta)}{\partial x(c_C,t_1)} & \dots & \frac{\partial f(x|\theta)}{\partial x(c_C,t_T)} \end{bmatrix} \quad (1)$$

The results displayed in Fig.2a and Fig. 2b show the average absolute gradient across all the *target* samples of a single cross validation test data: the warm colors correspond to high gradient values, indicating that the model is more sensitive to change in these input features. We can see the sensitivity of the CNN model spreads across the recording relatively evenly as opposed to the LSTM-CNN which is focused around the 250ms and 450ms time-stamps.

Table 4: Accuracy when introducing temporal noise. The best results are boldfaced when the differences are statistically significant.

Noise	CNN	LSTM_CNN	LDA
-120	0.058	0.044	0.016
-80	0.275	0.299	0.016
-40	**0.825**	**0.864**	**0.565**
40	**0.848**	**0.896**	**0.608**
80	0.335	0.390	0.260
120	0.042	0.042	0.059

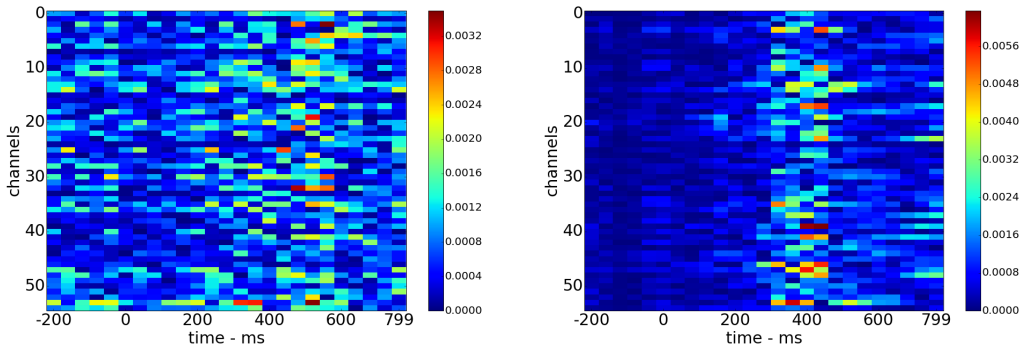
DISCUSSION

In this work we examined using LSTM neural networks for the task of the BCI task of P300 speller. Despite its temporal nature, no version of LSTM investigated in this work has shown a significant advantage compared to the CNN model suggested by [5]. LSTM results improved with large amounts of data from multiple subjects, and superior results are obntained with a combined CNN-LSTM model; moreover, we have shown that this combined model is significantly more robust to temporal noise in the stimuli onset. We also show that the sensitivity of the LSTM based model is much more focused on the area between 250ms to 450ms than CNN based model, which is in line with our expectation from

 DOI: 10.3217/978-3-85125-533-1-87

Table 3: Accuracy when training and testing on different subjects.

subject	LDA all except one	CNN all except one	CNN all except one fine tune	SMALL LSTM-CNN all except one	SMALL LSTM-CNN all except one fine tune
fat	0.94	1.00	1.00	0.98	1.00
gcb	0.43	0.83	0.91	0.86	0.92
gcc	0.79	0.98	0.98	0.95	0.97
gcd	0.66	0.80	0.99	0.83	0.97
gcf	0.68	0.89	0.98	0.79	0.98
gcg	0.52	0.81	0.94	0.77	0.90
gch	0.87	0.97	0.97	0.97	0.99
iay	0.48	0.69	0.98	0.67	0.97
icn	0.44	0.58	0.92	0.61	0.95
icr	0.63	0.81	1.00	0.89	1.00
pia	0.77	0.87	0.96	0.91	0.97
mean	0.65	0.84	0.97	0.84	0.97



(a) CNN

(b) LSTM-CNN

Figure 2: Average gradient in target samples, comparing CNN and LSTM-CNN.

the P300 ERP. To conclude – in the dataset we have explored a simple algorithm such as LDA performed extremely well when trained and tested on the same subject, but additional experiments involving cross-subject training and temporal noise expose the possible advantages of 'deep' networks, and especially the LSTM-CNN combined method.

REFERENCES

[1] L. Acqualagna and B. Blankertz., "Gaze-independent BCI-spelling using rapid serial visual presentation (rsvp)," *NeuroImage*, vol. 124, pp. 901–908, 2013.

[2] P. Bashivan, "Learning representations from EEG with deep recurrent-convolutional neural networks." *arXiv*, vol. 1511.06448, 2015.

[3] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[4] B. Blankertz, "Single-trial analysis and classification of erp components—a tutorial," *NeuroImage*, vol. 50, pp. 814–825, 2011.

[5] H. Cecotti and A. Graser, "Convolutional neural networks for p300 detection with application to brain-computer interfaces," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, pp. 433–445, 2011.

[6] F. Cincotti, A. Scipione, A. Timperi, D. Mattia, A. Marciani, J. Millan, S. Salinari, L. Bianchi, and F. Bablioni, "Comparison of different feature classifiers for brain computer interfaces," in *Neural Engineering, 2003. Conference Proceedings. First International IEEE EMBS Conference on*. IEEE, 2003, pp. 645–647.

[7] P. Davidson, R. Jones, and M. Peiris, "Detecting behavioural microsleeps using eeg and lstm recur-

 DOI: 10.3217/978-3-85125-533-1-87

rent neural networks," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc*, vol. 27, 2005, pp. 5754–5757.

[8] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Neural Computation*, vol. 70, pp. 510–523, 1988.

[9] A. Graves, *Supervised Sequence Labelling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 5–13.

[10] A. Graves, M. Liwicki, H. Bunke, J. Schmidhuber, and S. Fernández, "Unconstrained on-line handwriting recognition with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2008, pp. 577–584.

[11] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[12] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[13] S. Hochreiter and J. Schmidhuber., "Long short-term memory," *arXiv*, vol. 9.8, pp. 1735–1780, 1997.

[14] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[16] Y. LeCun, "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998.

[17] X.-D. Li, J. K. Ho, and T. W. Chow, "Approximation of dynamical time-variant systems by continuous-time recurrent neural networks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 10, pp. 656–660, 2005.

[18] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for eeg-based brain–computer interfaces," *Journal of neural engineering*, vol. 4, no. 2, p. R1, 2007.

[19] R. Manor and A. B. Geva, "Convolutional neural network for multi-category rapid serial visual presentationBCI," *Frontiers in computational neuroscience*, vol. 9, 2015.

[20] B. Obermaier, C. Guger, C. Neuper, and G. Pfurtscheller, "Hidden markov models for online classification of single trial eeg data," *Pattern recognition letters*, vol. 22, no. 12, pp. 1299–1309, 2001.

[21] A. Rakotomamonjy and V. Guigue, "BCI competition iii: dataset ii-ensemble of svms for BCI p300 speller," *IEEE transactions on biomedical engineering*, vol. 55, no. 3, pp. 1147–1154, 2008.

[22] M. Soleymani, S. Asghari-Esfeden, M. Pantic, and Y. Fu, "Continuous emotion detection using eeg signals and facial expressions," in *2014 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2014, pp. 1–6.

[23] S. Solhjoo, A. M. Nasrabadi, and M. R. H. Golpayegani, "Classification of chaotic signals using hmm classifiers: Eeg-based mental task classification," in *Signal Processing Conference, 2005 13th European*. IEEE, 2005, pp. 1–4.

[24] I. Sutskever, G. E. Hinton, and G. W. Taylor, "The recurrent temporal restricted boltzmann machine," in *Advances in Neural Information Processing Systems*, 2009, pp. 1601–1608.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[26] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, 2012.

[27] 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." in *ICML*, vol. 14, 2015, pp. 77–81.

[28] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4694–4702.