A BoW-equivalent Recurrent Neural Network for Action Recognition

Alexander Richard
richard@iai.uni-bonn.de
Juergen Gall
gall@iai.uni-bonn.de

Institute of Computer Science III
University of Bonn
Bonn, Germany

Bag-of-words (BoW) models are widely used in the field of computer vision and find application in texture and object classification, object discovery, and action recognition. In the traditional BoW model, kMeans or a Gaussian mixture model is used to generate a visual vocabulary based on which a histogram of visual words is computed. This histogram is then used for classification, *e.g.* in combination with a support vector machine (SVM). In our paper, we show an equivalent formulation of BoW as a recurrent neural network that allows to train the visual words discriminatively and directly on video level rather than on frame level only. We apply our method to four action recognition benchmarks as well as two small image recognition datasets and show its superiority over the classical model and some sparse coding methods such as LLC [3] and ScSPM [4].

In the classical approach, a set $\mathbf{x}$ of T feature vectors is quantized into a histogram of K visual words $v_1, \dots, v_K$,

$$\mathcal{H}(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} h(x_t), \quad h(x_t) = \begin{pmatrix} p(v_1|x_t) \\ \vdots \\ p(v_K|x_t) \end{pmatrix}, \tag{1}$$

where $p(v_k|x_t)$ is 1 if v_k is the visual word being closest to x_t and 0 otherwise. As an alternative to this hard assignment, soft assignment is frequently used, *i.e.* $p(v_k|x_t)$ is defined as a posterior distribution of a Gaussian mixture model or the kMeans model. Utilizing that kMeans is a special case of Gaussian mixture models, where the visual word prior is uniform and all variances are set to 1, we formulate an equivalent representation as a single-layer neural network with softmax output,

$$p_{\text{NN}}(v_k|x) := \text{softmax}_k(\mathbf{W}^\intercal x + b), \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{D\times K}$ is a weight matrix and $b \in \mathbb{R}^K$ the bias. Setting these to

$$\mathbf{W} = (v_1 \dots v_K), \quad b = -\frac{1}{2}(v_1^\intercal v_1 \dots v_K^\intercal v_K)^\intercal \tag{3}$$

results in a neural network that generates the same posterior distribution as kMeans with soft assignment.

In order to realize the summation over all x_t that is required in Equation (1), we add a recurrent layer with the unit matrix as weights to the network. The normalization over the sequence length is realized by the activation function

$$\sigma_t(z) = \begin{cases} z & \text{if } t < T, \\ \frac{1}{T}z & \text{if } t = T. \end{cases} \tag{4}$$

An additional softmax layer is finally added to model a posterior distribution of the action classes given the video, *c.f.* Figure 1.

Standard neural network training algorithms can now be applied to optimize the weights and biases in the model. The softmax layer we added on top of the network is only a linear classifier. In the classical BoW model, usually a SVM with non-linear kernel is applied. Thus, we use the softmax classifier only during training in order to obtain visual words that are optimized

1. discriminatively, *i.e.* separating the different classes as good as possible, and
2. on video level, *i.e.* the visual vocabulary is learned to best differentiate between collections of feature vectors that form the video rather than between single feature vectors or video frames only.

Once the network is trained, the output layer is discarded. The neural network histograms of an input sequence $\mathbf{x} = x_1, \dots, x_T$ are then used in combination with an SVM after application of a non-linear kernel.

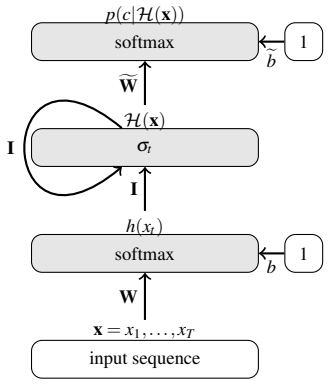

Figure 1: Neural network encoding the BoW model. The output layer is discarded after training and the histograms from the recurrent layer are used for classification in combination with an SVM.

Method	HMDB-51	UCF101
Traditional models		
Improved DT + bag of words	52.2%	73.3%
Improved DT + fisher vectors [2]	57.2%	85.9%
Improved DT + LLC	50.8%	71.9%
Neural networks		
Composite LSTM [1]	44.0%	75.8%
Ours	**54.0%**	**76.9%**

Table 1: Comparison of our model to published results on HMDB-51 and UCF101.

Note that the result of the neural network training can be transformed back into a traditional kMeans model with non-uniform prior. The corresponding visual words and the prior can be computed as

$$v_k = (\mathbf{W}_{1,k} \dots \mathbf{W}_{D,k})^\intercal, \tag{5}$$

$$p_{\text{NN}}(v_k) = \frac{\exp\left(b_k + \frac{1}{2}v_k^\intercal v_k\right)}{\sum_{\tilde{k}} \exp\left(b_{\tilde{k}} + \frac{1}{2}v_{\tilde{k}}^\intercal v_{\tilde{k}}\right)}. \tag{6}$$

Using improved dense trajectories [2] as video features, we conduct experiments on four different action recognition benchmarks. We observe a consistent improvement between two and five percent compared to the traditional BoW model and also outperform other methods that learn video representations discriminatively, such as LLC or composite LSTMs [1], see Table 1. An evaluation of our method on two image datasets led to similar results.

[1] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. *arXiv preprint arXiv:1502.04681*, 2015.

[2] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Int. Conf. on Computer Vision*, pages 3551–3558, 2013.

[3] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 3360–3367, 2010.

[4] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 1794–1801, 2009.