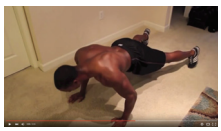
RNN Fisher Vectors for Action Recognition and Image Annotation

Guy Lev , Gil Sadeh, Benjamin Klein, Lior Wolf

ECCV'16 EUROPEAN CONFERENCE ON COMPUTER VISION

Video Action Recognition

Surfing



Pushups

Image-Sentence Retrieval

Image Annotation

A boy is riding a bicycle
Two girls are playing soccer
A man is playing guitar
A man is walking down the street

Image Search

A man is playing guitar

Sequence Representation

- Video: a sequence of frames / sub-volumes (blocks of 16-frames)
- Sentence: a sequence of words

The challenge: Represent a variable-length sequence by a fixed-length vector.

Fisher Vector (FV):
- A gradient-based fixed-length vector representation for a multiset of vectors → Insensitive to order of elements
- Traditional FV [47] assumes GMM distribution of the data

The idea: Replace the GMM probabilistic model by a Recurrent Neural Network (RNN) model.

RNN-FV: Fisher Vector that is sensitive to order of elements

Element Representation

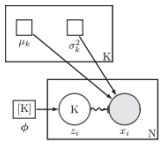
Video case
- Represent frame using VGG [13] CNN; or alternatively:
- Represent sub-volume using C3D [14] CNN

Sentence case
- Represent word using word2vec [6] embedding

Traditional Fisher Vector

Assumes GMM distribution of the element vectors:
- K multivariate Gaussians with diagonal covariance matrix
- Parameters of the model are:
  - $\tau_1, \dots, \tau_K \in R$ prior probabilities
  - $\mu_1, \dots, \mu_K \in R^D$ means
  - $\sigma_1, \dots, \sigma_K \in R^D$ variances

The parameters are learned by the EM algorithm



For a given GMM, and a set of vectors $X = \{x_1, \dots, x_n\} \in R^D$
Consider the log-likelihood of X given the parameters of the model $\lambda = \{\mu, \sigma, \tau\}$:

$$L(\lambda|X) = \sum_{i=1}^{n} \log(p(x_i|\lambda))$$

For each parameter $t \in \lambda$, we can compute: $\frac{\partial L(\lambda|X)}{\partial t}$
The vector $FV(X)$ is the gradient:

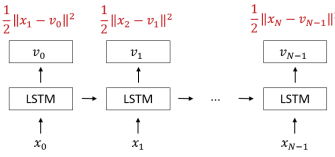
$$\nabla_\lambda L(\lambda|X)$$

- This representation does not take the elements ordering into account
- To overcome this insensitivity to ordering, we replace the GMM by an **RNN** model

RNN Training

We train an RNN to predict the next element in a sequence, given the previous ones.

Original sequence: $(x_1, \dots, x_N)$ $(x_i \in R^D)$
Input sequence: $X = (x_0, \dots, x_{N-1})$ $(x_0 = x_{start})$
Target sequence: $Y = (x_1, \dots, x_N)$
Loss function: $Loss(y, v) = \frac{1}{2}\|y - v\|^2$



RNN Feature Extraction

Given a new sequence $X = (x_1, \dots, x_N)$, feed it to the RNN.
At time step i, the output of the RNN is v_i
and the likelihood of a vector $x \in R^D$ can be seen as:

$$p(x|x_0, \dots, x_i) = (2\pi)^{-D/2} \exp\left(-\frac{1}{2}\|x - v_i\|^2\right)$$

The likelihood of the correct next word x_{i+1} is:

$$p(x_{i+1}|x_0, \dots, x_i) = (2\pi)^{-D/2} \exp\left(-\frac{1}{2}\|x_{i+1} - v_i\|^2\right)$$

The likelihood of the entire sequence X is:

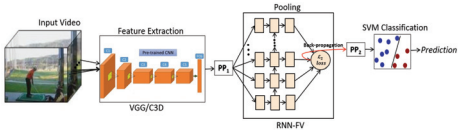
$$p(X) = \prod_{i=0}^{N-1} p(x_{i+1}|x_0, \dots, x_i)$$

The negative log-likelihood of X:

$$L(X) = \frac{ND}{2}\log(2\pi) + \frac{1}{2}\sum_{i=0}^{N-1}\|x_{i+1} - v_i\|^2$$

$L(X)$ equals (up to an additive constant) the loss we would get by feeding X to the RNN
→ $FV(X) = \nabla_W L(X)$ can be computed by backpropagation

Video Action Recognition Pipeline

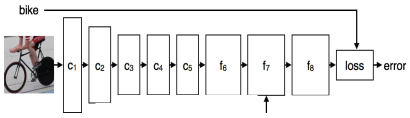


Post processing:
- PP_1: PCA/CCA dimension reduction and L2 normalization
- PP_2: PCA dimension reduction followed by power and L2 normalization

Image-Sentence Retrieval Pipeline

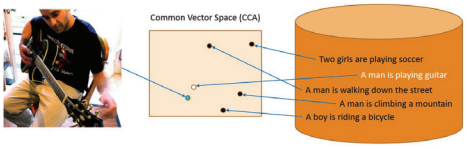
Sentence representation: RNN-FV
- RNN may be trained on external corpus (e.g., Wikipedia)
  → Generic model

Image representation: VGG [13] Convolutional Neural Network (CNN)



Bringing images and sentences into a common domain:
- We apply *Canonical Correlation Analysis (CCA)* on the training set.
- The CCA returns two projection matrices W_{image}, $W_{sentence}$ which map the image vectors and sentence vectors to a common vector space.

Retrieval / Ranking
- Given an image vector, rank the sentences according to cosine similarity in the common vector space
- Given a sentence vector, rank the images the same way



Video Action Recognition - Results

Accuracy results on the HMDB51 and UCF101 datasets

Method	HMDB51	UCF101
idt [49]	57.2	85.9
idt + high-D encoding [53]	61.1	87.9
Two-Stream CNN (2 nets) [28]	59.4	88
Multi-Skip feature stacking [54]	65.4	89.1
C3D (1 net) [14]	-	82.3
C3D (3 nets) [14]	-	85.2
C3D (3 nets) + idt [14]	-	90.4
TDD (2 nets) [5]	63.2	90.3
TDD (2 nets) + idt [5]	65.9	91.5
Stacked FV [2]	56.21	-
Stacked FV + idt [2]	66.78	-
RNN-FV (C3D + VGG-CCA)	54.33	88.01
RNN-FV (C3D + VGG-CCA) + idt	**67.71**	**94.08**

RNN-FV vs. GMM

HMDB51

Method	GMM-FV	RNN-FV
VGG PCA	36.80	45.62
VGG CCA	39.61	46.14
C3d	45.82	52.88

UCF101

Method	GMM-FV	RNN-FV
VGG PCA	76.53	79.29
VGG CCA	76.84	79.49
C3d	80.04	82.33

Image-Sentence Retrieval - Results

Recall@1 results on the flickr8k and flickr30k datasets

	Flickr8k		Flickr30k	
Method	Image Annotation	Image Search	Image Annotation	Image Search
m-CNN [41]	24.8	20.3	33.6	26.2
NIC [40]	20.0	19.0	17.0	17.0
SC-NLM [37]	18.0	12.5	23.0	16.8
m-RNN [55]	14.5	11.5	35.4	22.8
SDT-RNN [33]	6.0	6.6	9.6	8.9
DVSA [39]	16.5	11.8	22.2	15.2
GMM FV [8]	28.4	20.6	33.0	23.9
GMM+HGLMM FV [8]	31.0	21.3	35.0	25.1
RNN-FV (Wikipedia-trained)	29.3	19.8	32.9	23.9
RNN-FV (CCA)	30.9	20.7	33.6	24.5
RNN-FV (Ens)	29.9	22.4	34.7	26.2
RNN-FV (Ens) + [8]	**31.6**	**23.2**	**35.6**	**27.4**

- Model trained on Wikipedia sentences performs well
  → Demonstrates generalization capability

References

[2] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. ECCV, 2014
[5] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. arXiv, 2015
[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. NIPS, 2013
[8] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. CVPR, 2015
[13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. ICLR, 2015
[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. arXiv, 2014.
[28] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. NIPS, 2014.
[33] R. Socher, Q. Le, C. Manning, and A. Ng. Grounded compositional semantics for finding and describing images with sentences. NIPS, 2013
[37] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. TACL, 2015.
[39] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. Technical report, Stanford University, 2014
[40] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. arXiv, 2014
[41] L.Ma, Z. Lu, L. Shang, and H. Li. Multimodal convolutional neural networks for matching image and sentence. ICCV, 2015
[47] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. CVPR, 2007
[49] H. Wang and C. Schmid. Action recognition with improved trajectories. ICCV, 2013
[53] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. arXiv, 2014.
[54] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj. Beyond Gaussian pyramid: Multi-skip feature stacking for action recognition. arXiv, 2014
[55] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. arXiv, 2014