ChaLearn Looking at People 2015 - Track 2: Cultural Event Recognition

September 15, 2015

1 Team details

- Team name

  VIPL-ICT-CAS

- Team members

Mengyi Liu (mengyi.liu@vipl.ict.ac.cn)

Xin Liu (xin.liu@vipl.ict.ac.cn)

Yan Li (yan.li@vipl.ict.ac.cn)

Shiguang Shan (sgshan@ict.ac.cn)

Xilin Chen (xlchen@ict.ac.cn)

- Affiliation

Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS

2 Contribution details

- Title of the contribution

  Exploiting Global and Local Information with Convolutional Neural Networks for Cultural Event Recognition

- General method description

  Our method is based on a combination of visual features extracted from deep convolutional neural networks. Specifically, we investigate two off-the-shelf architectures, VGGNet [3] and GoogLeNet [4], and adapt them to our task by performing event-specific fine-tuning on both global and local images. For "global" scheme, we take the whole image as input; while for "local" scheme, we first generate a batch of region proposals in each image and take these local regions as inputs. In recognition stage,

we employ two kinds of linear classifiers, Logistic Regression (LR) [2] and Linear Discriminant Analysis (LDA) [1], on image features from different deep models and fusing their decision scores for final results. (Please refer to section 3-5 for details.)

- Final scores (Validation)

  *Model type format:

  [Networks architectures] [Dataset] ... [Dataset] [Region (optional)]

Table 1: Performance based on VGGNet.

Features	mAP	
	Classifier (LR)	Classifier (LDA)
VGG16ImageNet	0.639	0.648
VGG16ImageNetRegion	0.707	0.697
VGG16ImageNetEvents	0.735	0.741
VGG19ImageNet	0.626	0.640
VGG19ImageNetRegion	0.709	0.695
VGG19ImageNetEvents	0.728	0.734

Table 2: Performance based on GoogLeNet.

Features	mAP	
	Classifier (LR)	Classifier (LDA)
GoogLeNetPlaces	0.505	0.416
GoogLeNetPlacesEvents	0.689	0.708
GoogLeNetImageNet	0.551	0.537
GoogLeNetImageNetEvents	0.723	0.739
GoogLeNetImageNetEventsRegion	0.805	0.804
GoogLeNetLoss1ImageNetEventsRegion	0.753	0.758
GoogLeNetLoss2ImageNetEventsRegion	0.788	0.793

Table 3: Performance based on fusion.

Features	mAP		
	LR	LDA	LR+LDA
GoogLeNetImageNetEventsRegion +GoogLeNetLoss1ImageNetEventsRegion +GoogLeNetLoss2ImageNetEventsRegion	0.813	0.804	0.824
+VGG16ImageNetEvents +VGG19ImageNetEvents	0.833	0.822	0.840
+VGG16ImageNetRegion +VGG19ImageNetRegion	0.841	0.821	0.849

*Region option: Region proposals generated by selective search [5] results filtering out regions whose a) width and height is less than 20% of the original image; b) width/height ratio is greater than 2.0 or less than 0.5.

3 Data Preprocessing

- "Global" scheme

  For a single image, if width $>$ height, we resize the image to keep its height as 256 pixels and resample it as left, middle, and right parts with the size of 256×256; otherwise, if width $<$ height, we resize the image to keep its width as 256 pixels, and resample it as top, middle, and bottom parts, also with the size of 256×256. For each image, we combine the feature vectors of these three parts by mean-pooling to obtain the final representation.

- "Local" scheme

  For a single image, we generate a batch of (about 125 per image in our experiments) region proposals using selective search by filtering out the results whose a) width and height is less than 20% of the original image; b) width/height ratio is greater than 2.0 or less than 0.5. All the sub-regions are then resized to 256×256 directly. For each image, we combine the feature vectors of all these regions by mean-pooling to obtain the final representation.

4 Classification details

We employ two kinds of linear classifiers, Logistic Regression (LR) and Linear Discriminant Analysis (LDA), on image features extracted from different deep models and fusing their decision scores for final results. For LR, we use the LibLinear package [2] with the parameter "$-s$ 0 $-c$ 1". For LDA, we first conduct PCA for dimension reduction. Specifically, we preserve 3,000 dimensions for VGGNet features and 1,000 dimensions for GoogLeNet features. The final LDA dimension is set as $\#category - 1 = 99$.

5 Global Method Description

- Feature extraction stage

  In our final submission, two architectures, VGGNet and GoogLeNet are employed for feature extraction. Both of them are pre-trained on ImageNet database. To adapt the models to our task, we conduct fine-tuning using the training set of event images provided by the challenge. Specifically, for VGGNet, we fine-tune the networks using $42,996$ ($\#train \times 3$) global images with 30K iterations, which takes about 1 day with Tesla K40

GPU. The output values of the last 4096-dimension fc layer are served as the final image representation. For GoogLeNet, we fine-tune the networks using $1,794,988$ region proposals with 100K iterations, which takes about 2 days with Tesla K40 GPU. The output values of 1024-dimension $pool5-7\times7$ layer are served as the final image presentation.

- Classification stage

  For classification, we summarize the training/testing expended time of both validation and final testing stage, regarding to different deep models and different classifiers. (Note that all data are obtained using one PC with 2.20GHz and 4G RAM.)

Table 4: Computational time in validation stage (#train=14332, #val=5704).

Features	LR		LDA	
	train	test	train	test
GoogLeNet	214.39s	0.82s	7.80s	10.11s
VGGNet	379.25s	1.12s	146.53s	10.37s

Table 5: Computational time in test stage (#train+#val=20036, #test=8669).

Features	LR		LDA	
	train	test	train	test
GoogLeNet	424.02s	1.17s	9.08s	32.48s
VGGNet	587.86s	1.65s	146.53s	37.92s

References

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *PAMI*, 19(7):711–720, 1997.

[2] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[5] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.