Better Exploiting OS-CNNs for Better Event Recognition in Images

Limin Wang, Zhe Wang, Sheng Guo, Yu Qiao

Shenzhen Institutes of Advanced Technology, CAS, China

December 12, 2015

Outline

1. Introduction
2. OS-CNNs Revisited
3. Exploring OS-CNNs
4. Experiments
5. Conclusions

Outline

Introduction

Figure: Examples of cultural event recognition dataset.

- Event recognition in still images is very important for image understanding, just like object and scene recognition.
- Event is a complex concept and relevant to many other factors, including objects, human poses, human garments and scene categories.

Motivations

- Object, scene, and event are three highly related concepts in high-level computer vision research.
- **As event is highly relevant with object and scene, transferring effective representations learned for object and scene recognition will be a reasonable choice. (our OS-CNN work)**
- **Both global and local representations of CNNs will help in event recognition and are complementary to each other. (our TDD work)**

L. Wang, Z. Wang, W. Du, and Y. Qiao *Object-scene convolutional neural networks for event recognition in images*, in CVPR ChaLearn Workshop, 2015.

L. Wang, Y. Qiao, and X. Tang *Action recognition with trajectory-pooled deep-convolutional descriptors*, in CVPR, 2015.

Outline

Overview

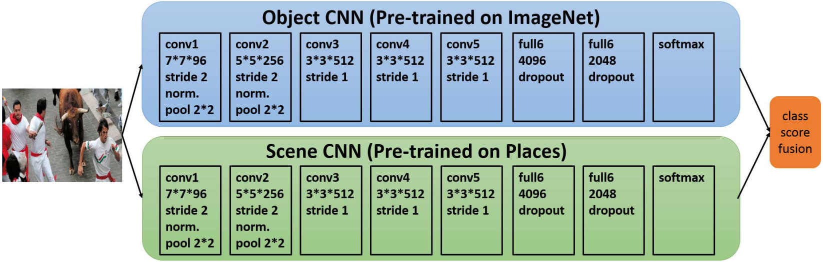

Figure: The architecture of Object-Scene Convolutional Neural Network (OS-CNN) for event recognition.

L. Wang, Z. Wang, W. Du, and Y. Qiao *Object-scene convolutional neural networks for event recognition in images*, in CVPR ChaLearn Workshop, 2015.

Object-Scene Convolutional Neural Networks are composed of two nets

- **Object nets**: capturing useful information of objects to help event recognition.
- We build object nets based on recent advances on object recognition and pre-train it on the ImageNet dataset.
- **Scene nets**: extracting scene information to assist event recognition.
- We construct scene nets with the help of recent works on scene recognition and pre-train it on the Places dataset.

Based on previous analysis, event is highly relevant with object and scene. Thus, we combine the recognition scores of both object and scene nets:

$$s(\mathbf{I}) = \alpha_o s_o(\mathbf{I}) + \alpha_s s_s(\mathbf{I}).$$

Implementation Details

- **Network structure**: we choose VGGNet-19 as our investigation structure [1].
- **Learning policy**: pre-train OS-CNNs with ImageNet-VGGNet models [1] and Places205-VGGNet models [2] + fine tuning.
- **Data augmentations**: we use common data augmentation techniques, such as corner crop, scale jittering, and horizontal flipping.
- **Speed up**: we design a Multi-GPU extension version of Caffe toolbox, that is publicly available [3].

K. Simonyan, and A. Zisserman *Very deep convolutional networks for large-scale image recognition*, in ICLR, 2015.

L. Wang, S. Guo, W. Huang, and Y. Qiao *Places205-VGGNet models for scene recognition*, in arXiv 1508.01667.

L. Wang, Y. Xiong, Z. Wang, and Y. Qiao *Towards good practices for very deep two-stream ConvNets*, in arXiv 1507.02159.

Outline

Scenario 1: OS-CNN Predictions

- In this scenario, we directly use the outputs (softmax layer) of OS-CNNs as final prediction results.

$$s_{os}(\mathbf{I}) = \alpha_o s_o(\mathbf{I}) + \alpha_s s_s(\mathbf{I}),$$

- $s_o(\mathbf{I})$ and $s_s(\mathbf{I})$ are the prediction scores of object nets and scene nets, α_o and α_s are their fusion weights.

- In this scenario, we treat OS-CNNs as generic feature extractors and extract the **global representation** of an image region.
- In this case, we only use the pre-trained models without fine-tuning.
- Specifically, we use the activations of **fully connected layers** as follows:

$$\phi_{os}^{p}(\mathbf{I}) = [\beta_o \phi_o^{p}(\mathbf{I}), \quad \beta_s \phi_s^{p}(\mathbf{I})],$$

- $\phi_o^p(\mathbf{I})$ and $\phi_s^p(\mathbf{I})$ are the CNN activations from pre-trained object nets and scene nets, β_o and β_s are the fusion weights.

Scenario 3: OS-CNN Global Representations (pre-training + fine tuning)

- In this scenario, We consider fine-tuning the OS-CNNs on the event recognition dataset and the resulted image representations become dataset-specific.
- After fine-tuning process, we obtain the following global representation with the fine-tuned OS-CNNs:

$$\phi_{os}^{f}(\mathbf{I}) = [\beta_o \phi_o^f(\mathbf{I}), \quad \beta_s \phi_s^f(\mathbf{I})],$$

- $\phi_o^f(\mathbf{I})$ and $\phi_s^f(\mathbf{I})$ are the CNN activations from the fine-tuned object nets and scene nets, β_o and β_s are the fusion weights.

Scenario 4: OS-CNN Local Representations (pre-training + fine tuning)

- We consider exploring the activations of **convolutional layers** and we call them as **local representations** of OS-CNNs.
- After extracting OS-CNN local representations, we use *channel normalization* and *spatial normalization* to pre-process them into transformed convolutional feature maps $\widetilde{C}(\mathbf{I}) \in \mathbb{R}^{n \times n \times c}$.
- The normalized CNN activation $\widetilde{C}(\mathbf{I})(x, y, :) \in \mathbb{R}^{c}$ at each postion is called as the *Transformed Deep-convolutional Descriptor* (TDD).
- Finally, we employ Fisher vector to encode these TDDs into a global representation.

L. Wang, Y. Qiao, and X. Tang *Action recognition with trajectory-pooled deep-convolutional descriptors*, in CVPR, 2015.

Outline

Experiment Setup

- The challenge dataset contains 100 event classes (99 event classes + 1 background) and it is divided into three parts: (i) development data (14,332 images), (ii) validation data (5,704 images), (iii) evaluation data (8669 images)
- As we can not access the label of evaluation data, we mainly train our models on the development data and report the results on the validation data.
- For final evaluation, we merge the development data and validation data into a single training dataset and re-train our OS-CNN models on this new dataset.
- In our exploration experiments, we report our results evaluated as AP value for each class and mAP value for all classes.

Experiment Results

	Object nets	Scene nets	OS-CNNs
Scenario 1			
softmax	73.1%	71.2%	75.6%
Scenario 2			
fc7	67.2%	63.4%	69.1%
Scenario 3			
fc6	80.6%	76.8%	81.7%
fc7	81.4%	78.1%	82.3%
Scenario 4			
conv5-1	77.6%	76.6%	78.9%
conv5-2	78.6%	76.2%	79.6%
conv5-3	79.4%	76.1%	80.2%
conv5-4	78.4%	75.6%	79.7%
Fusion			
conv5-3+fc7	82.5%	79.3%	83.2%

Experiment Results (cont'd)

- Object nets outperform scene nets and the combination of them improves recognition performance.
- Combining fine tuned features with linear SVM classifier (scenario 3) is able to obtain better performance than direct using the softmax output of CNNs (scenario 1).
- Comparing fine-tuned features (scenario 3) with pre-trained features (scenario 2), we may conclude that fine tuning on the target dataset is very useful.
- Global representations (scenario 3) is better than local ones (scenario 4) and the combination of them further boots the recognition performance.

Experiment Results (cont'd)

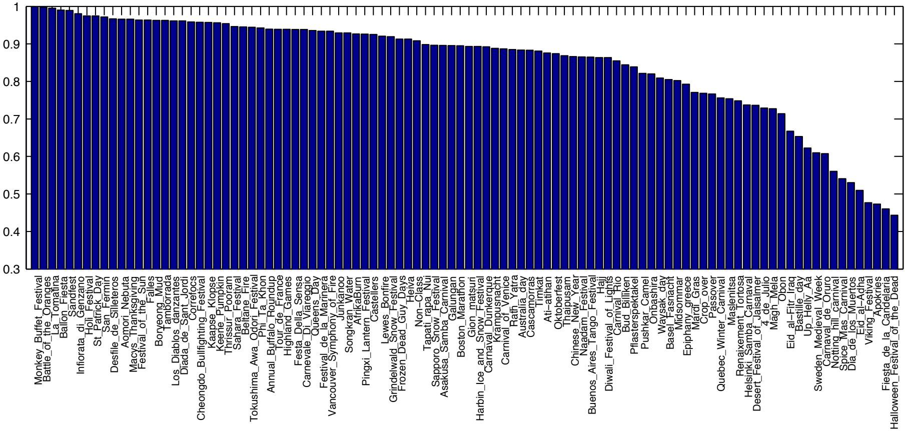

Figure: Per-class AP value of combining OS-CNN global and local representations on the validation data.

Experiment Results (cont'd)

Figure: Examples of images that our method succeeds and fails in top-1 evaluation.

Challenge Results

Rank	Team	Score
1	VIPL-ICT-CAS	85.4%
2	FV	85.1%
3	**MMLAB (ours)**	84.7%
4	NU&C	82.4%
5	CVL_ETHZ	79.8%
6	SSTK	77.0%
7	MIPAL_SUN	76.3%
8	ESB	75.8%
9	Sungbin Choi	62.4%
10	UPC-STP	58.8%

Table: Comparison the performance of our submission with those of other teams. Our team secures the third place in the ICCV ChaLearn LAP challenge 2015.

Outline

Conclusions

- We have presented a new architecture for event recognition, called *object-scene convolutional neural networks* (OS-CNN), by capturing effective information from the perspectives of object and scene.
- From our experimental results, object nets outperform scene nets on event recognition, and the combination of them further improve performance.
- We comprehensively study four scenarios to better exploit OS-CNNs for better cultural event recognition.
- Global representations (fully connected layers) is a bit better than local representations (convolutional layers) and the combination of them further boots the recognition performance.

code and model coming soon at https://wanglimin.github.io