DeCAF: a Deep Convolutional Activation Feature for Generic Visual Recognition

Features Representation’s challenge

PROBLEM

- performance with conventional visual representations (*flat feature representations*) has been impressive but has likely plateaued in recent years

SOLUTION

- discover effective representations that capture salient semantics for a given task
- deep architectures should be able to do this
A little bit of History

- Deep CNN has a long history in computer vision
  - supervised back-propagation networks to perform digit recognition [LeCun et al., 1989]

- Recently CNN have achieved competition-winning numbers on large benchmark dataset
  - convolutional network proposed by Krizhevsky (2012)
  - dataset consisting of more than one million images (ImageNet) [Berg et al., 2012]

- Learning from related tasks has also a long history in machine learning [Caruana, 1997 - Argyriou et al., 2006]

- In computer Vision forming a representation based on sets of trained classifiers on related tasks has recently show to be effective [Torresani et al., 2010 - Li et al., 2010]

PROBLEM

- Transfer learning using deep representation bad in unsupervised setting
  - limited with relatively small datasets (CIFAR and MNIST)
  - modest success on larger datasets [Le et al., 2012]
Why Deep Models

- deep or layered compositional architectures should be able to capture salient aspects of a given domain [Krizhevsky *NIPS 2012*][Singh *ECCV 2012*]

- perform better than traditional hand-engineered representations in many domains
  - especially where good features has not already been engineered [Le *CVPR 2011*]

- recently applied to large-scale visual recognition tasks
  - recently outperformed all known methods on a large scale recognition challenge
  - performs extremely well in domains with large amounts of training data

**HOWEVER**

- with limited training data, fully-supervised deep architectures generally overfit
- many conventional visual recognition challenges have tasks with few training examples
Idea

‣ investigate a deep architecture
  - representations are learned on a set of related problems
  - applied to new tasks which have too few training examples

‣ model considered as a deep architecture for transfer learning
  - based on a supervised pre-training phase
  - new visual features “DeCAF” defined by convolutional network weights

WHY

‣ empirical validation
  - that generic visual feature based on a CNN weights trained on ImageNet outperforms conventional visual representations

WITH

‣ Caltech-101 (Object recognition dataset [Fei-Fei et al., 2004])
‣ Office (Domain adaptation dataset [Saenko et al., 2010])
‣ Caltech-UCSD (Birds fine-grained recognition dataset [Welinder et al., 2010])
‣ SUN-397 (Scene recognition dataset [Xiao et al., 2010])
Approach

- Train a Deep convolutional model in a fully supervised setting using Krizhevsky method
  - state-of-the-art method
  - large scale dataset for training (ImageNet)
- Extract various features from the network
- Evaluate the efficacy of these features on generic vision tasks

TWO IMPORTANT QUESTIONS

- Do features extracted from the CNN generalize the other datasets?
- How does performance vary with network depth?

FEEDBACK

- qualitatively and quantitatively via visualizations of semantic clusters
- experimental comparison to current baselines
Adopted Network

- Deep CNN architecture proposed by Krizhevsky et al. (2012)
  - values propagated through 5 convolutional layers (with pooling and ReLU)
  - 3 fully-connected layers to determine final neuron activities
  - won ImageNet Large Scale Visual recognition Challenge 2012 [Berg et al., 2012]
  - top-1 validation error rate of 40.7%

- Follow architecture and training protocol with two differences
  - input 256 x 256 images rather than 224 x 224 images
  - no data augmentation trick (e.g. adding random multiples of the p.c of the RGB)
To gain insight into the semantic capacity of DeCAF features

Comparison with GIST features [Oliva & Torralba, 2001] and LLC features [Wang et al., 2010]

Use of t-SNE algorithm [van der Maaten & Hilton, 2008]
- find 2-dimensional embedding of the high-dimensional feature space
- plot as a points colored depending on their semantic category

Use of ILSVRC-2012 validation set to avoid overfitting (150,000 photographs, collected from flickr and other search engines)

Use of SUN-397 dataset to evaluate how dataset bias affects results
Take the activations of $n$ hidden layer of the CNN as a feature $\text{DeCAF}_n$. 

- (a) LLC
- (b) GIST
- (c) $\text{DeCAF}_1$
- (d) $\text{DeCAF}_6$
Experimental Comparison Feedback

- Experimental results evaluating DeCAF on multiple standard computer vision benchmarks
- Not evaluation of features from any earlier layers in the CNN
  - do not contain rich semantic representation
- Results on multiple datasets to evaluate the strength of DeCAF for
  - basic object recognition (Caltech-101)
  - domain adaptation (Office)
  - fine-grained recognition (Caltech-UCSD)
  - scene recognition (SUN-397)
- Together represent much of the contemporary visual recognition spectrum
Object Recognition

- Evaluation also of a regularization technique called “dropout” [Hilton et al., 2012]

- Classifier trained on random set of 30 samples per class and tested on the rest

- Results compared with current state-of-the-art on this benchmark [Yang et al. 2009]
  - combination of 5 traditional hand-engineered image features

- Compared also with the two-layers convolutional network of Jarret et al (2009)
  - to demonstrate the importance of the depth of the network used for this features

<table>
<thead>
<tr>
<th></th>
<th>DeCAF5</th>
<th>DeCAF6</th>
<th>DeCAF7</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogReg</td>
<td>63.29 ± 6.6</td>
<td>84.30 ± 1.6</td>
<td>84.87 ± 0.6</td>
</tr>
<tr>
<td>LogReg with Dropout</td>
<td>-</td>
<td>86.08 ± 0.8</td>
<td>85.68 ± 0.6</td>
</tr>
<tr>
<td>SVM</td>
<td>77.12 ± 1.1</td>
<td>84.77 ± 1.2</td>
<td>83.24 ± 1.2</td>
</tr>
<tr>
<td>SVM with Dropout</td>
<td>-</td>
<td>86.91 ± 0.7</td>
<td>85.51 ± 0.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mean Accuracy per Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogReg DeCAF6 w/ Dropout</td>
<td>[Graph of accuracy vs. Num Train per Category]</td>
</tr>
<tr>
<td>SVM DeCAF6 w/ Dropout</td>
<td></td>
</tr>
<tr>
<td>Yang et al. (2009)</td>
<td>84.3</td>
</tr>
<tr>
<td>Jarrett et al. (2009)</td>
<td>65.5</td>
</tr>
</tbody>
</table>
Domain Adaptation 1/2

- Particular dataset used with three domains
  - *Amazon*: images taken from amazon.com
  - *Webcam* and *Dslr*: images taken in office environment using a webcam or SLR camera

<table>
<thead>
<tr>
<th></th>
<th>Amazon -&gt; Webcam</th>
<th>Dslr -&gt; Webcam</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SURF</td>
<td>DeCAF$_6$</td>
</tr>
<tr>
<td>Logistic Reg. (S)</td>
<td>9.63 ± 1.4</td>
<td>48.58 ± 1.3</td>
</tr>
<tr>
<td>SVM (S)</td>
<td>11.05 ± 2.3</td>
<td>52.22 ± 1.7</td>
</tr>
<tr>
<td>Logistic Reg. (T)</td>
<td>24.33 ± 2.1</td>
<td>72.56 ± 2.1</td>
</tr>
<tr>
<td>SVM (T)</td>
<td>51.05 ± 2.0</td>
<td>78.26 ± 2.6</td>
</tr>
<tr>
<td>Logistic Reg. (ST)</td>
<td>19.89 ± 1.7</td>
<td>75.30 ± 2.0</td>
</tr>
<tr>
<td>SVM (ST)</td>
<td>23.19 ± 3.5</td>
<td>80.66 ± 2.3</td>
</tr>
<tr>
<td>Daume III (2007)</td>
<td>40.26 ± 1.1</td>
<td>82.14 ± 1.9</td>
</tr>
<tr>
<td>Hoffman et al. (2013)</td>
<td>37.66 ± 2.2</td>
<td>80.06 ± 2.7</td>
</tr>
<tr>
<td>Gong et al. (2012)</td>
<td>39.80 ± 2.3</td>
<td>75.21 ± 1.2</td>
</tr>
<tr>
<td>Chopra et al. (2013)</td>
<td></td>
<td>58.85</td>
</tr>
</tbody>
</table>

- Multi-class accuracy averaged across 5 train/test splits for domain shift
- Three ways of training
  - with only source data (S)
  - with only target data (T)
  - with source and target data (ST)
Domain Adaptation 2/2

- DeCAF robust to resolution changes (t-SNE algorithm)

- DeCAF provides better category clustering than SURF

- DeCAF clusters same category instances across domains
Fine-Grained Recognition (subcategory recognition)

- Caltech-UCSD birds dataset [Welinder et al., 2010]
- Performance comparison against several state-of-the-art baselines

Two approaches
- First adopt ImageNet-like pipeline, DeCAF6 and a multi-class logistic regression
- Second adopt deformable part descriptors (DPD) method [Zhang et al., 2013]

Outperforms also POOF with the best accuracy performed in the literature
Scene Recognition

- SUN-397 large-scale scene recognition database [Xiao et al., 2010]
- Goal: classify the scene of the entire image

- Used 50 training samples and 50 test samples per class
  - Results averaged across 5 splits of 50 training images and 50 test images
  - Top-performing method selected by cross-validation

<table>
<thead>
<tr>
<th></th>
<th>DeCAF₆</th>
<th>DeCAF₇</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogReg</td>
<td>40.94 ± 0.3</td>
<td>40.84 ± 0.3</td>
</tr>
<tr>
<td>SVM</td>
<td>39.36 ± 0.3</td>
<td>40.66 ± 0.3</td>
</tr>
</tbody>
</table>

- Outperforms Xiao et al. (2010), the current state-of-the-art method

- DeCAF demonstrate
  - the ability to generalize to other tasks
  - representational power as compared to traditional hand-engineered features
Discussion

DONE

▷ Analysis of the use of deep features applied in semi-supervised multi-task framework

DEMONSTRATIONS

▷ Using a large labeled object database to train a deep convolutional architecture
  - is possible to learn features with representational power and generalization ability
  - is possible to perform good semantic visual discrimination tasks with linear classifiers
  - outperform current state-of-the-art approaches

VISUAL RESULTS

▷ Demonstrate the generality and semantic knowledge implicit in DeCAF features
  ▷ Showing that features tend to cluster images into interesting semantic categories

NUMERICAL RESULTS

▷ DeCAF frameworks can improve the performance of a wide variety of existing method
  ▷ Improving across a spectrum of visual recognition tasks
References


Le, Q., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G., Dean, J., and Ng, A. Building high-level features using large scale unsupervised learning. In ICML, 2012.


Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. JMLR, 9, 2008.


Some links


› Caffe (DeCAF improvement): [http://caffe.berkeleyvision.org/](http://caffe.berkeleyvision.org/)

› Alex Krizhevsky convolutional neural network: [https://code.google.com/p/cuda-convnet/](https://code.google.com/p/cuda-convnet/)


› t-SNE: [http://homepage.tudelft.nl/19j49/t-SNE.html](http://homepage.tudelft.nl/19j49/t-SNE.html)