DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition
Autonomous Car
Real-time object recognition
Object classification

Convolutional Neural Network

- Car
- Traffic Light
- Street Sign
- ...

Training
CNN for Object Recognition

Feature extraction

Classification

Source: [2]
Feature extraction

Source: https://developer.apple.com
From features to object classes

High-level features:
- Shape of a car
- Road marking
- Face with eyes and ears
- Cat skin

Classes:
- Cat
- Car
- …

Source: [2]
Visualization of high dimensional feature space

- Visualization with t-SNE algorithm [4]

Source: [1]
Repurpose Features from CNN

Convolutional Neural Network → Object class

Learned Features

Convolutional Neural Network
Classification with small training dataset

Freeze trained convolution kernels

Target database

High-level features

Logistic Regression

Classify new database

Source: [2]
Experiments: Are features transferrable to solve new tasks?

- Train AlexNet [2] on ILSVRC 2012 object recognition dataset
- Reuse extracted features for new tasks:
  - Experiment #1: Basic Object Recognition
  - Experiment #2: Domain Adaption
  - Experiment #3: Fine-grained recognition
  - Experiment #4: Scene recognition
 Experiment #1: Basic object recognition

- Classify new objects on new dataset (Caltech-101 dataset)
- 2.6% better than state-of-art

<table>
<thead>
<tr>
<th>Model</th>
<th>DeCAF$_5$</th>
<th>DeCAF$_6$</th>
<th>DeCAF$_7$</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>
<tr>
<td>Yang et al. (2009)</td>
<td>84.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jarrett et al. (2009)</td>
<td>65.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mean Accuracy per Category vs Num Train per Category
Experiment #2: Domain adaption

- Train object recognition in different surrounding, only few labeled data in target domain available
- Office dataset

<table>
<thead>
<tr>
<th></th>
<th>Amazon → Webcam</th>
<th>Dslr → 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>19.80 ± 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>58.85</td>
<td>78.21</td>
</tr>
</tbody>
</table>

Source: [1]
Experiment #3: Subcategory recognition

- Caltech-UCSD birds dataset
- 8% better than state-of-art

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeCAF&lt;sub&gt;6&lt;/sub&gt;</td>
<td>58.75</td>
</tr>
<tr>
<td>DPD + DeCAF&lt;sub&gt;6&lt;/sub&gt;</td>
<td>64.96</td>
</tr>
<tr>
<td>DPD (Zhang et al., 2013)</td>
<td>50.98</td>
</tr>
<tr>
<td>POOF (Berg &amp; Belhumeur, 2013)</td>
<td>56.78</td>
</tr>
</tbody>
</table>

*Table 2. Accuracy on the Caltech-UCSD bird dataset.*
Experiment #4: Scene recognition

- Classes like abbey, diner, mosque, stadium
- SUN-397 dataset
- >2% better than state-of-art

<table>
<thead>
<tr>
<th></th>
<th>DeCAF$_6$</th>
<th>DeCAF$_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogReg</td>
<td><strong>40.94 ± 0.3</strong></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>
<tr>
<td><strong>Xiao et al. (2010)</strong></td>
<td><strong>38.0</strong></td>
<td><strong>38.0</strong></td>
</tr>
</tbody>
</table>

Source: [1]
Conclusions

▪ Extract features from ILSVRC dataset to solve new classification tasks
▪ State-of-the-art performance in 4 different tasks
▪ CNN features are generic enough to solve completely new problems
▪ Bigger datasets yield better accuracy
▪ Release of DeCAF (predecessor of Caffe)
Conclusions cont.
Conclusions cont.

- Challenges:
  - Find labeled data
  - Training time of CNN

- Convolutional Neural Network

- Car
- Traffic Light
- Street Sign
- …
Q&A
References


Computing time of forward propagation

Source: [1] and [2]