Cross-Domain Self-supervised Multi-task Feature Learning using Synthetic Imagery.

To Appear at CVPR 2018

Zhongzheng (Jason) Ren, Yong Jae Lee
UC Davis
Cross-Domain Self-supervised Multi-task Feature Learning using Synthetic Imagery.
Predicting Labels from Data

Supervised training

Data $x$
ImageNet images

Learned feature hierarchy

Label $y$
ImageNet labels
Predicting Data from Data

Supervised training

Unsupervised/Self-supervised training

Learned feature hierarchy

Label $y$

ImageNet images

Learned feature hierarchy

$x_0$

$x_1$

$x_0$

$x_1$

ImageNet labels
Cross-Domain Self-supervised Multi-task Feature Learning using Synthetic Imagery.
Autoencoders
Hinton & Salakhutdinov.
Science 2006.

Denoising Autoencoders
Vincent et al. ICML 2008.

Audio
Owens et al. CVPR 2016, ECCV 2016

Colorization
Zhang et al. ECCV 2016.

These are single task

Egomotion
Agrawal et al. ICCV 2015
Jayaraman et al. ICCV 2015

Context
Doersch et al. ICCV 2015
Pathak et al. CVPR 2016

Video
Wang et al. ICCV 2015
Multi-task could help

• **Lessos learnt from other tasks:**
  • RCNN Family: bbox regression + Classification + (Inst. Segmentation)
  • 3D geometry: Depth + Surface Normal
  • ...

• **For Feature learning:**

Doersch *et al.* ICCV 2017

Pinto *et al.* ECCV 2016
Multi-task could help

- Lessons learnt from other tasks:
  - RCNN Family: bbox regression + Classification + (Inst. Segmentation)
  - 3D geometry: Depth + Surface Normal

- For Feature learning:

Doersch et al. ICCV 2017

Pinto et al. ECCV 2016
Where to get free and multiple Annotations?

Cross-Domain Self-supervised Multi-task Feature Learning using Synthetic Imagery.
The Advantage of Game Engine

• It's become more and more realistic and will only become better

• Tremendous resources that already on the market

• Full freedom to change anything

• Renderring with GT is easy

• Easy to scale
The Architecture

Three pixel-level tasks using Encoder-Decoder arch.
Cross-Domain Self-supervised Multi-task Feature Learning using Synthetic Imagery.
To minimize the domain gap
How does Domain Adaptation work?

Algorithm 1 Multi-task Adversarial Domain Adaptation

Input: Synthetic images $X$, real images $Y$, max iteration $T$
Output: Domain adapted base network $B$
1: for $t = 1$ to $T$ do
2: Sample a batch of synthetic images $\{x_i\}$
3: Sample a batch of real images $\{y_j\}$
4: Extract feature for each image: $z_{x_i} = B(x_i), z_{y_j} = B(y_j)$
5: Keep $D$ frozen, update $B, H$ through $L_{BH}(\phi_B, \phi_H | z_x)$
6: Keep $B$ frozen, update $D$ through $L_D(\phi_D | z_x, z_y)$

- Inspired on Generative Adversarial Network (GAN)
- $B$ tries to generate features which can fool $D$ (close to natural) which $D$ also learns to discriminate
Experiments
Qualitative
Nearest Neighbor

Query
Random weights
Ours w/o Domain Adaptation

Ours
ImageNet
Experiments
Qualitative

conv1 filters

Ours

ImageNet
Experiments Qualitative Prediction

Synthetic RGB

Depth Pred.

Depth GT

Surface normal Pred.

Surface normal GT

Instance contour Pred.

Instance contour GT
Table 1. Transfer learning results on PASCAL VOC 2007 classification and VOC 2007 and 2012 detection. We report the best numbers for each method reported in [35, 76, 52].
<table>
<thead>
<tr>
<th>method</th>
<th>conv1</th>
<th>conv2</th>
<th>conv3</th>
<th>conv4</th>
<th>conv5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageNet [36]</td>
<td>19.3</td>
<td>36.3</td>
<td>44.2</td>
<td>48.3</td>
<td>50.5</td>
</tr>
<tr>
<td>Gaussian</td>
<td>11.6</td>
<td>17.1</td>
<td>16.9</td>
<td>16.3</td>
<td>14.1</td>
</tr>
<tr>
<td>Krähenbühl et al. [35]</td>
<td>17.5</td>
<td>23.0</td>
<td>24.5</td>
<td>23.2</td>
<td>20.6</td>
</tr>
<tr>
<td>context [14]</td>
<td>16.2</td>
<td>23.3</td>
<td>30.2</td>
<td>31.7</td>
<td>29.6</td>
</tr>
<tr>
<td>BiGAN [16]</td>
<td>17.7</td>
<td>24.5</td>
<td>31.0</td>
<td>29.9</td>
<td>28.0</td>
</tr>
<tr>
<td>context-encoder [55]</td>
<td>14.1</td>
<td>20.7</td>
<td>21.0</td>
<td>19.8</td>
<td>15.5</td>
</tr>
<tr>
<td>colorization [75]</td>
<td>12.5</td>
<td>24.5</td>
<td>30.4</td>
<td>31.5</td>
<td>30.3</td>
</tr>
<tr>
<td>jigsaw [51]</td>
<td><strong>18.2</strong></td>
<td>28.8</td>
<td>34.0</td>
<td>33.9</td>
<td>27.1</td>
</tr>
<tr>
<td>splitbrain [76]</td>
<td>17.7</td>
<td>29.3</td>
<td><strong>35.4</strong></td>
<td><strong>35.2</strong></td>
<td><strong>32.8</strong></td>
</tr>
<tr>
<td>counting [52]</td>
<td>18.0</td>
<td><strong>30.6</strong></td>
<td>34.3</td>
<td>32.5</td>
<td>25.7</td>
</tr>
</tbody>
</table>

Table 2. Transfer learning results on ImageNet [13]. We freeze the weights of our model and train a linear classifier for ImageNet classification [13]. Our model is trained purely on synthetic data while all other methods are trained on ImageNet [13] (without labels). Despite the domain gap, our model still learns useful features for image classification.
Experiments
Quantitative
(Transfer learning)

<table>
<thead>
<tr>
<th>Task</th>
<th>Adaptation</th>
<th>#data</th>
<th>07-C</th>
<th>07-D</th>
<th>12-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge</td>
<td>-</td>
<td>0.5M</td>
<td>63.9</td>
<td>46.9</td>
<td>44.8</td>
</tr>
<tr>
<td>Depth</td>
<td>-</td>
<td>0.5M</td>
<td>61.9</td>
<td>48.9</td>
<td>45.8</td>
</tr>
<tr>
<td>Surf.</td>
<td>-</td>
<td>0.5M</td>
<td>65.3</td>
<td>48.2</td>
<td>45.4</td>
</tr>
<tr>
<td>3 tasks</td>
<td></td>
<td>0.5M</td>
<td>65.6</td>
<td>51.3</td>
<td>47.2</td>
</tr>
<tr>
<td>3 tasks</td>
<td>conv1</td>
<td>0.5M</td>
<td>61.9</td>
<td>48.7</td>
<td>46</td>
</tr>
<tr>
<td>3 tasks</td>
<td>conv4</td>
<td>0.5M</td>
<td>63.4</td>
<td>49.5</td>
<td>46.3</td>
</tr>
<tr>
<td>3 tasks</td>
<td><strong>conv5</strong></td>
<td>0.5M</td>
<td><strong>67.4</strong></td>
<td><strong>52.0</strong></td>
<td><strong>49.2</strong></td>
</tr>
<tr>
<td>3 tasks</td>
<td>conv6</td>
<td>0.5M</td>
<td>66.9</td>
<td>51.5</td>
<td>48.2</td>
</tr>
<tr>
<td>3 tasks</td>
<td>conv5 Bi-fool</td>
<td>0.5M</td>
<td>66.2</td>
<td>51.3</td>
<td>48.5</td>
</tr>
<tr>
<td>3 tasks</td>
<td><strong>conv5</strong></td>
<td><strong>1.5M</strong></td>
<td><strong>68.0</strong></td>
<td><strong>52.6</strong></td>
<td><strong>50.0</strong></td>
</tr>
</tbody>
</table>

Table 3. Ablation study results. We evaluate the impact of multi-task learning, feature space domain adaptation, and amount of data on transfer learning. All of these factors contribute together to make our model learn transferable visual features from large-scale synthetic data.
Experiments
Quantitative
Cross-Domain

NYUD

<table>
<thead>
<tr>
<th>GT</th>
<th>Methods</th>
<th>Lower the better</th>
<th>Higher the better</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
<td>Median</td>
</tr>
<tr>
<td>[17]</td>
<td>Zhang et al. [78]</td>
<td>22.1</td>
<td>14.8</td>
</tr>
<tr>
<td>[17]</td>
<td>Ours</td>
<td>21.9</td>
<td>14.6</td>
</tr>
<tr>
<td>[38]</td>
<td>Wang et al. [71]</td>
<td>26.0</td>
<td>18.0</td>
</tr>
<tr>
<td>[38]</td>
<td>Ours</td>
<td>23.8</td>
<td>16.2</td>
</tr>
</tbody>
</table>

Table 4. Surface normal estimation on the NYUD [50] test set.
Thanks!


* **Code coming soon at** [jason718.github.io](https://jason718.github.io)

* **Self-Supervised Learning Resources:**