Multi-Task Self-Supervised Visual Learning

Sikai Zhong
March 4, 2018

COMPUTER SCIENCE
1. Introduction

2. Self-supervised Tasks

3. Architectures

4. Experiments
Introduction
Self-supervised Learning

- It does **not** have manual labeling;
- The objective is measured by the performance of the task;
It combines the following four tasks to boost performance.

- Relative Position;
- Colorization;
- "Exemplar" task
- motion segmentation;
• Tasks learn at different rates;
• A naive combination of self-supervision tasks will conflict;
Self-supervised Tasks
Relative Position

- Sampling two patches from a single image randomly
- Pairs of patches are taken from adjacent grid points (eight-way softmax classification)
Colorization

- Given a grayscale image and predict the color of every pixel;[2]
- The color are vectors quantized into 313 different categories;
- 313-way softmax classification for every region of the image;
1. Create pseudo-classes, where each class was generated by taking a patch from a single image and augmenting it via translation, rotation, scaling, and color shifts;[1]

2. Randomly sample two patches $x_1$ and $x_2$ from the same pseudo-class, and a third patch $x_3$ from a different pseudo-class.

3. Train the network with a loss of the form

$$\max(D(f(x_1); f(x_2))D(f(x_1); f(x_3)) + M; 0),$$
Given a single frame of video, it asks the network to classify which pixels will move in subsequent frames.
Architectures
Figure 1: The structure of our multi-task network. It is based on ResNet-101, with block 3 having 23 residual units. a) Naive shared-trunk approach, where each head is attached to the output of block 3. b) the lasso architecture, where each head receives a linear combination of unit outputs within block 3, weighted by the matrix $\alpha$, which is trained to be sparse.
Different tasks require different features;

- Some information is useful to imageNet classification but useless to object detection;
- Some tasks need only image patches but some tasks need the entire image;
Separating features via Lasso

- Each task has a set of coefficients, one for each of the 23 candidate layers in block 3 (Figure.5);
- Lasso(L1) is used to encourage the matrix to be **sparse**;
- Sparse matrix will encourage the network to concentrate all of the information required by a single task into a small number of layers;
• Each self-supervised task pre-process its data differently, so the low-level image statistics are often very different across tasks;
• We replace relative position’s preprocessing with the same preprocessing used for colorization;
• Images are converted to Lab, and the a and b channels are discarded.
• L channels replicate the L channels 3 times so that the network can be evaluated on color images;
Head for Relative Position

- Input: a batch of patches;
- Running ResNet-v2-101 at a stride of 8 with a dilated convolution (most block 3 convolutions produce outputs at stride 16);
- Header has two more residual units. The first has an output with 1024 channels, a bottleneck with 128 channels, and a stride of 2; the second has an output size of 512 channels, bottleneck with 128 channels, and stride 2.
- 3 fully-connected residual units is used to process the flatten feature map;
• the input images are 256*256;
• Running ResNet-v2-101 at a stride of 8 with a dilated convolution;
• Header has two more standard convolution layers with a ReLU nonlinearity (one has 2*2 kernel with stride 1, the other has 1*1 kernel with stride 1, they both have 4096 output channels);
• The last layer is a 1*1 convolution with stride 1 and 313 outputs channels;
• Input: Images are resized to 256*256 and sample patches that are 96*96.
• Running ResNet-v2-101 at a stride of 8 with a dilated convolution;
• Header has two residual units, the first with an output with 1024 channels, a bottleneck with 128 channels, and a stride of 2; the second has an output size of 512 channels, bottleneck with 128 channels, and stride 2. The feature map is used directly to compute the distances needed for the loss.
• Input: Image are resized to 240*320;
• Running ResNet-v2-101 at a stride of 8 with a dilated convolution;
• header have two 1*1 conv layers each with dimension 4096, followed by another 1*1 conv layer which produces a single value, which is treated as a logit and used a per-pixel classification.
Training

**Figure 2:** Distributed training setup. Several GPU machines are allocated for each task, and gradients from each task are synchronized and aggregated with separate RMSProp optimizers.
Experiments
Datasets

- **ImageNet**: used in relative position, colorization, exemplar;
- **SoundNet**: used in motion segmentation;
Comparison of performance for different self-supervised methods over time

Figure 3: Comparison of performance for different self-supervised methods over time. X-axis is compute time on the self-supervised task (2.4K GPU hours per tick). Random Init shows performance with no pre-training.
Comparison of performance for different multi-task self-supervised methods over time

**Figure 4:** Comparison of performance for different multi-task self-supervised methods over time. X-axis is compute time on the self-supervised task (2.4K GPU hours per tick). Random Init shows performance with no pre-training.
Mediated combination of self-supervision tasks

Figure 5: Comparison of performance with and without the lasso technique for factorizing representations, for a network trained on all four self-supervised tasks for 16.8K GPU-hours.

<table>
<thead>
<tr>
<th>Net structure</th>
<th>ImageNet</th>
<th>PASCAL</th>
<th>NYU</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Lasso</td>
<td>69.30</td>
<td>70.53</td>
<td>79.25</td>
</tr>
<tr>
<td>Eval Only Lasso</td>
<td>70.18</td>
<td>68.86</td>
<td>79.41</td>
</tr>
<tr>
<td>Pre-train Only Lasso</td>
<td>68.09</td>
<td>68.49</td>
<td>78.96</td>
</tr>
<tr>
<td>Pre-train &amp; Eval Lasso</td>
<td>69.44</td>
<td>68.98</td>
<td>79.45</td>
</tr>
</tbody>
</table>
C. Doersch and A. Zisserman.  
**Multi-task self-supervised visual learning.**  

R. Zhang, P. Isola, and A. A. Efros.  
**Colorful image colorization.**  