Self-Supervised Learning of Pretext-Invariant Representations

Presenter: Rohan Nair

9/16/2021
Motivation and Main Problem

Modern vision systems learn semantics from large datasets
Predefined semantics tasks have long tails and do not model the problem well
❖ These models are brittle and not very robust, require defining the semantics in pretraining settings
❖ Poor generalizability means applications will not lend themselves to unseen situations well
One solution is to transform image data, and have the model predict properties about the known transformation
❖ Does not learn an invariant representation of the image, model learns a covariance
Motivation and Main Problem

Key contributions of this paper:

❖ Invariant representations are much more useful than covariant ones for image tasks
❖ Want to learn representations that are similar to transformed images and dissimilar from other images + their transformations
❖ Benchmark invariant representations with other covariant techniques (Jigsaw)
Problem Setting

Problem Formulation

- Let our image dataset $D = \{I_1, \ldots, I_{|D|}\}$ with $I_n \in \mathbb{R}^{H \times W \times 3}$
- Let $T$ be our set of transformations.
  - Focus on Jigsaw in this paper (slice up image and rearrange the patches)
- Goal: construct neural network $\Phi_\theta(\cdot)$ s.t. $\Phi_\theta(I) = v_I$ is invariant to transformation $t \in T$
- Invariance loss function: Empirical risk minimization
  - $p(T)$ is a distribution over the transformation
  - $L$ is a similarity function between 2 representations

$$\ell_{inv}(\theta; D) = \mathbb{E}_{t \sim p(T)} \left[ \frac{1}{|D|} \sum_{I \in D} L(v_I, v_{I^t}) \right]$$
Problem Setting

Problem Formulation

- Author contrast their loss against other papers:
  - $z(t)$ is a function that measures some properties of $t$
  - Encourages model to learn some information about the transformation itself
    - Results in covariant representations

$$
\ell_{co}(\theta; D) = \mathbb{E}_{t \sim p(T)} \left[ \frac{1}{|D|} \sum_{I \in D} L_{co}(v_I, z(t)) \right]
$$
Problem Setting

Problem Formulation

- Defining $L$ concretely:
  - Use Noise Contrastive Estimator (NCE) with distribution $h$
  - NCE models the probability that $(I, I^T)$ come from distribution $h$
    - $s$ is the cosine similarity function
- Finalized Loss Function:
  - Feed convolutional representation $v$ through “head” function $f$ and $g$
  - This encourages the model to learn representations of $I$ to be close to transformations $I^T$ but far away from $I'$ or transformations of $I'$

$$h(v_I, v_{I^t}) = \frac{\exp \left( \frac{s(v_I, v_{I^t})}{\tau} \right)}{\exp \left( \frac{s(v_I, v_{I^t})}{\tau} \right) + \sum_{I' \in \mathcal{D}_N} \exp \left( \frac{s(v_{I'}, v_{I^t})}{\tau} \right)}.$$ 

$$L_{NCE} (I, I^t) = - \log [h (f(v_I), g(v_{I^t}))] - \sum_{I' \in \mathcal{D}_N} \log [1 - h (g(v_{I'}), f(v_{I^t}))]$$
Context / Related Work / Limitations of Prior Work

❖ Compare their approach primarily to the model from Jigsaw (Nozoori and Favaro 2016)

❖ Previous works have learned representations of images covariant with their transformations
  ○ This is undesirable for semantic learning tasks
  ○ Images are transformed in a way that defeats the semantic understanding portion of the task
Context / Related Work / Limitations of Prior Work

Summary of other approaches:

- **2 Highly related works:**
  - NPID: Maximally distance out learned features using NCE, doesn’t use any transformations
  - Jigsaw: Predict permutation of jigsaw pieces, does not optimize distancing image representations

- **Reconstruction based approaches:**
  - Autoencoders
  - GANs
  - Sparse Coding

- **Image-based Pretext Tasks:**
  - Affine Transformation
  - Colorization
  - Orientation Prediction
Proposed Approach / Algorithm / Method
Proposed Approach / Algorithm / Method

● Use a ResNet-50 as the convolutional model

● f and g are 128 dimensional representations
  ○ f is obtained by extracting res5 features, average pooling, and a linear projection
  ○ g is obtained by:
    ■ extracting nine patches from image I
    ■ computing an image representation for each patch separately by extracting activations from the res5 layer of the ResNet-50 and average pooling the activations
    ■ applying a linear projection to 128 dimensions
    ■ concatenating the patch representations in random order and apply a second linear projection to 128 dimensions
Proposed Approach / Algorithm / Method

- Practical limitation: NCE requires a large number of negative samples
  - A large number of samples is infeasible to compute while keeping batch size reasonably small
  - Solution: keep a memory bank of average representations of \( f(v) \)
    - Exponential moving average kept in a cache
    - Representations only computed on \( I \), not \( I^T \)
  - Final Loss function with memory bank:
    - First term is NCE from before with \( f(v) \) and \( f(v') \) swapped with \( m_i \) and \( m_i' \)
    - Second term encourages \( f(v_i) \) to be similar to memory representation \( m_i \) and for \( f(v_i) \) and \( f(v_i') \) to be dissimilar

\[
L(I, I^t) = \lambda L_{NCE}(m_I, g(v_{1t})) + (1 - \lambda) L_{NCE}(m_I, f(v_I)).
\]
Proposed Approach / Algorithm / Method
Experimental Setup

PIRL evaluated on the task of transfer learning

❖ Pretrain on large corpus of image data
❖ Learn generalized representations of Images
❖ Transfer to domain with limited data available

Dataset used to evaluate was ImageNet
❖ 1.28 M Images
Experimental Setup

3 Downstream Tasks Evaluated

❖ Object Detection (VOC07)
❖ Image Classification with Linear Models (ImageNet, VOC07, Places205, and iNaturalist2018)
❖ Semi-supervised Image Classification (ImageNet)

1 Other Pretraining Domain evaluated:
❖ Pretraining on Uncurated Data (YFCC)
## Experimental Results

### Task 1: Object Detection

<table>
<thead>
<tr>
<th>Method</th>
<th>Network</th>
<th>AP&lt;sup&gt;all&lt;/sup&gt;</th>
<th>AP&lt;sup&gt;50&lt;/sup&gt;</th>
<th>AP&lt;sup&gt;75&lt;/sup&gt;</th>
<th>ΔAP&lt;sup&gt;75&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>R-50</td>
<td>52.6</td>
<td>81.1</td>
<td>57.4</td>
<td>=0.0</td>
</tr>
<tr>
<td>Jigsaw [19]</td>
<td>R-50</td>
<td>48.9</td>
<td>75.1</td>
<td>52.9</td>
<td>-4.5</td>
</tr>
<tr>
<td>Rotation [19]</td>
<td>R-50</td>
<td>46.3</td>
<td>72.5</td>
<td>49.3</td>
<td>-8.1</td>
</tr>
<tr>
<td>NPID++ [72]</td>
<td>R-50</td>
<td>52.3</td>
<td>79.1</td>
<td>56.9</td>
<td>-0.5</td>
</tr>
<tr>
<td>PIRL (ours)</td>
<td>R-50</td>
<td><strong>54.0</strong></td>
<td><strong>80.7</strong></td>
<td><strong>59.7</strong></td>
<td>+2.3</td>
</tr>
<tr>
<td>CPC-Big [26]</td>
<td>R-101</td>
<td>–</td>
<td>70.6*</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>MoCo [24]</td>
<td>R-50</td>
<td>55.2*†</td>
<td>81.4*†</td>
<td>61.2*†</td>
<td></td>
</tr>
</tbody>
</table>
## Experimental Results

### Task 2: Image Classification

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
<th>Transfer Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ImageNet</td>
</tr>
<tr>
<td>Supervised</td>
<td>25.6M</td>
<td>75.9</td>
</tr>
<tr>
<td>Colorization [19]</td>
<td>25.6M</td>
<td>39.6</td>
</tr>
<tr>
<td>Rotation [18]</td>
<td>25.6M</td>
<td>48.9</td>
</tr>
<tr>
<td>NPID++ [72]</td>
<td>25.6M</td>
<td>59.0</td>
</tr>
<tr>
<td>MoCo [24]</td>
<td>25.6M</td>
<td>60.6</td>
</tr>
<tr>
<td>Jigsaw [19]</td>
<td>25.6M</td>
<td>45.7</td>
</tr>
<tr>
<td>PIRL (ours)</td>
<td>25.6M</td>
<td>63.6</td>
</tr>
</tbody>
</table>

Different architecture or evaluation setup:

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
<th>Transfer Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPID   [72]</td>
<td>25.6M</td>
<td>54.0</td>
</tr>
<tr>
<td>BigBiGAN [12]</td>
<td>25.6M</td>
<td>56.6</td>
</tr>
<tr>
<td>AET    [76]</td>
<td>61M</td>
<td>40.6</td>
</tr>
<tr>
<td>Rot.   [33]</td>
<td>61M</td>
<td>54.0</td>
</tr>
<tr>
<td>LA     [80]</td>
<td>25.6M</td>
<td>60.2(^\d)</td>
</tr>
<tr>
<td>CMC    [64]</td>
<td>51M</td>
<td>64.1</td>
</tr>
<tr>
<td>CPC    [51]</td>
<td>44.5M</td>
<td>48.7</td>
</tr>
<tr>
<td>CPC-Huge [26]</td>
<td>305M</td>
<td>61.0</td>
</tr>
<tr>
<td>BigBiGAN-Big [12]</td>
<td>86M</td>
<td>61.3</td>
</tr>
</tbody>
</table>
## Experimental Results

### Task 3: Semi Supervised Learning

<table>
<thead>
<tr>
<th>Method</th>
<th>Data fraction → Backbone</th>
<th>1% Top-5 Accuracy</th>
<th>10% Top-5 Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random initialization [72]</td>
<td>R-50</td>
<td>22.0</td>
<td>59.0</td>
</tr>
<tr>
<td>NPID [72]</td>
<td>R-50</td>
<td>39.2</td>
<td>77.4</td>
</tr>
<tr>
<td>Jigsaw [19]</td>
<td>R-50</td>
<td>45.3</td>
<td>79.3</td>
</tr>
<tr>
<td>NPID++ [72]</td>
<td>R-50</td>
<td>52.6</td>
<td>81.5</td>
</tr>
<tr>
<td>VAT + Ent Min. [20, 45]</td>
<td>R-50v2</td>
<td>47.0</td>
<td>83.4</td>
</tr>
<tr>
<td>$S^4$L Exemplar [75]</td>
<td>R-50v2</td>
<td>47.0</td>
<td>83.7</td>
</tr>
<tr>
<td>$S^4$L Rotation [75]</td>
<td>R-50v2</td>
<td>53.4</td>
<td>83.8</td>
</tr>
<tr>
<td>PIRL (ours)</td>
<td>R-50</td>
<td>57.2</td>
<td>83.8</td>
</tr>
<tr>
<td>Colorization [36]</td>
<td>R-152</td>
<td>29.8</td>
<td>62.0</td>
</tr>
<tr>
<td>CPC-Largest [26]</td>
<td>R-170 and R-11</td>
<td>64.0</td>
<td>84.9</td>
</tr>
</tbody>
</table>
## Experimental Results

### Unsupervised Pretraining Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>Transfer Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ImageNet</td>
</tr>
<tr>
<td>Jigsaw [19]</td>
<td>YFCC1M</td>
<td>–</td>
</tr>
<tr>
<td>DeepCluster [6, 7]</td>
<td>YFCC1M</td>
<td>34.1</td>
</tr>
<tr>
<td>PIRL (ours)</td>
<td>YFCC1M</td>
<td>57.8</td>
</tr>
<tr>
<td>Jigsaw [19]</td>
<td>YFCC100M</td>
<td>48.3</td>
</tr>
<tr>
<td>DeeperCluster [7]</td>
<td>YFCC100M</td>
<td>45.6</td>
</tr>
</tbody>
</table>
Discussion of Results

1-2 slides

What conclusions are drawn from the results by the authors?

❖ Quantitatively, PIRL outperforms all other similar methods
  ○ PIRL is also reasonably efficient with the number of parameters as compared to SOTA models
❖ However, supervised learning still performs the best

Are the stated conclusions fully backed by the results and references?

❖ Pretrain task performance is supported by these experiments
  ○ End task is a whole other metric + experiment, further analysis will need to be conducted to back up that this is a better pretrained model
Analysis on Model

Authors ran 4 analyses on model performance:

- Visualizing aggregate distances between representations of model
- Analyzing performance of several layers on image classification (testing against Jigsaw’s model)
- Setting lambda to different values in the loss function
- Increasing the number of patches to permute to demonstrate scale of transformations handled
- Performance improvement with increasing the number of negative samples
Ablation Results

![Graph showing Proportion of Samples vs $l_2$ distance between unit norm representations for PIRL and Jigsaw [19]]
Analysis Results
Analysis Results

![Graph showing Top-1 Accuracy vs. Relative weight of loss terms (\(\lambda\)) in Equation 5]

- **Graph Description:**
  - **X-axis:** Relative weight of loss terms (\(\lambda\)) in Equation 5
  - **Y-axis:** Top-1 Accuracy
  - **Legend:** ImageNet

---

CS391R: Robot Learning (Fall 2021)
Analysis Results

![Graph showing Analysis Results]

- mAP vs. # of permutations of patches
- PIRL vs. Jigsaw [19]
Analysis Results

Figure 8: Effect of varying the number of negative samples. Top-1 accuracy of linear classifiers trained to perform ImageNet classification using PIRL representations as a function of the number of negative samples, $N$. 
Critique / Limitations / Open Issues

Key Limitations

● The framework is not ideal for images that may be quite closely related semantically
  ○ Contrastive loss may be too strong

● No metrics provided on training speed

● Limited in scope in that they only really have one set of transformations
  ○ Future survey paper for performance on a larger set of transformations could provide even better results
Future Work for Paper / Reading

❖ Paper could be extended to other transformations

❖ Clustering based approaches for images that are visually very similar
  ○ Currently the model penalizes against all images that aren’t the original input
  ○ Perhaps other visually similar images need not be distanced
Summary

❖ We want better more robust representations for visual semantics
❖ Leads to more robust and generalizable models
❖ Prior work trains models that have covariant representations with transformations
❖ Want invariant representations
❖ This can lead to more robust pretrained vision models