Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey

Longlong Jing and Yingli Tian

Presented by: Qiaojun Feng
05/30/2019

Some slides from: Andrew Zisserman
Relevant codes: https://github.com/google/revisiting-self-supervised

Term Definition

• Human-annotated label:
Labels of data that are manually annotated by human workers.

Input: an image
Label:
• Bounding box coordinate
• Binary segmentation mask
Term Definition

- **Pseudo label**: Automatically generated labels based on data attributes for pretext tasks.

- **Pretext Task**: Pre-designed tasks for networks to solve, and visual features are learned by learning objective functions of pretext tasks.

Input: a color image
Pseudo input: a grayscale image
Pseudo label: a color image
Pretext task: predict pixel color

---

Term Definition

- **Supervised Learning**: Using data with fine-grained human-annotated labels to train networks.

- **Self-supervised Learning**: ConvNets are explicitly trained with automatically generated labels. Learned visual feature learning can be transferred to multiple different computer vision tasks.
Pretext Tasks

• The pretext tasks and pseudo labels share two common properties:

(1) Pseudo labels for the pretext task can be automatically generated based on the attributes of images or videos. (Without any human supervision.)

(2) Visual features of images or videos need to be captured by ConvNets to solve the pretext tasks. (Solving the task is equivalent to learning the feature.)

Pretext Tasks

Mainly focus on a) generation-based b) context-based and c) cross modal-based methods.
Generation-Based Methods

- Image inpainting
  A task of predicting arbitrary missing regions based on the rest of an image.
  - Pseudo input: image with missing regions \( f(I) \)
  - Pseudo label: origin image \( I \)

Given an image with a missing region (a), a human artist has no trouble inpainting it (b). Automatic inpainting using our context encoder trained with L2 reconstruction loss is shown in (c), and using both L2 and adversarial losses in (d).

Generation-Based Methods

- Image super-resolution (SR)
  A task of enhancing the resolution of images.
  From size \( W \times H \) to \( rW \times rH \) with \( r > 1 \).
  - Pseudo input: downsampled image \( I^{LR} = f(I^{HR}) \)
  - Pseudo label: origin image \( I^{HR} \)
Generation-Based Methods

• Image/Video colorization
A task of predicting a plausible color version of the photograph given a gray-scale photograph as input.
• Pseudo input: grayscale image $I_{gray} = f(I_c)$
• Pseudo label: origin image $I_c$

Generation-Based Methods

• Image/Video generation with GAN
  • Generator: map any latent vector sampled from latent space into an image/a video
  • Discriminator: distinguish whether the image/video from the real data distribution or generated data distribution.
• Pseudo label: origin image $I$
• The parameters of the discriminator can serve as the pre-trained model for other computer vision tasks.
Generation-Based Methods

• Video prediction
  A task of predicting future frame sequences based on a limited number of frames of a video.
  • Pseudo input: first $t$ frames $I_{1:t}$
  • Pseudo label: following $n$ frames $I_{t+1:t+n}$

Context-Based Methods

• Clustering
  A method of grouping sets of similar data in the same clusters.
  • Pseudo label: origin image $I$
Context-Based Methods

• Jigsaw Image Puzzle

The pretext task can be to predict the relative positions of two patches from same image, or to recognize the order of the shuffled a sequence of patches from same image.

(a) is an image with 9 sampled image patches, (b) is an example of shuffled image patches, and (c) shows the correct order of the sampled 9 patches.
Context-Based Methods

The context of full images can also be used as a supervision signal to design pretext tasks such as to recognize the rotating angles of the whole images.

Context-Based Methods

• Video frame order verification/recognition

Temporal order verification is to verify whether a sequence of input frames is in correct temporal order, while temporal order recognition is to recognize the order of a sequence of input frames.
Cross Modal-Based Methods

- Camera pose estimation (and depth estimation)

With ego-odometry measurements from ego-motor sensor signals (IMU) as supervision, do supervised learning

Without odometry, do self-supervised learning by predicting depth and ego-motion jointly.

Cross Modal-Based Methods

- Optical flow estimation

- RGB-Flow correspondence
Cross Modal-Based Methods

• Visual-Audio Correspondence

![Visual-Audio Correspondence Diagram]

• Video-Speech Synchronization

![Video-Speech Synchronization Diagram]

Evaluation

• Self-supervised pre-trained models are fine-tuned on downstream tasks such as
  • image classification
  • semantic segmentation
  • object detection
  • action recognition (video)

• The performance demonstrates the generalization ability of the learned features.
Evaluation

• Performance of image feature:
  Comparable to the supervised methods.

• Performance of video feature:
  Still much lower than the supervised models.
  • 3DConvNets usually have more parameters which lead to easily over-fitting
  • The temporal dimension of the video increase the complexity of learning.

• Current evaluation metric does not give insight what the network learned through the self-supervised pre-training.

Future Directions

• Learning from synthetic data:
  Use various simulators/game engines.

• Learning from multiple sensors:
  Driving dataset can have visual sensors together with odometry sensors.

• Learning from multiple pretext tasks
Motivation

• Autoencoder has bottleneck structure for forcing model abstraction.

• Context encoders (missing chunks and inpainting) has a “domain gap” between training (cropped image) and testing (full image).

• Colorization only uses grayscale, leaving the color information unused.

• How to train a representation:
  • Without explicit bottleneck constraint
  • Use all the input so that no domain gap between training/testing
Problem

- Raw input \( \mathbf{X} \in \mathbb{R}^{H \times W \times C} \)
- Splitted data \( \mathbf{X}_1 \in \mathbb{R}^{H \times W \times C_1}, \mathbf{X}_2 \in \mathbb{R}^{H \times W \times C_2} \)
- Prediction problem \( \hat{\mathbf{X}}_2 = \mathcal{F}(\mathbf{X}_1) \)
  Fit \( \mathcal{F} \) using a CNN. Suppose \( \mathcal{F}^l \) is the \( l \)-th layer
- Pretext task
  Minimize the loss function \( \ell(\mathcal{F}(\mathbf{X}_1), \mathbf{X}_2) \)
- Ultimate goal
  Find a good feature representation \( \mathcal{F}^l \ldots \mathcal{F}^1 \)

Approach
Approach

- Learn cross-channel encoders on opposite prediction problems
  \[ F_1 = \arg \min_{F_1} L1(F_1(X_1), X_2), \]
  \[ F_2 = \arg \min_{F_2} L2(F_2(X_2), X_1). \]

  ![Diagram](Image)

- By concatenating the representations layer-wise, \( F^l = \{ F^l_1, F^l_2 \} \), we achieve a representation \( F \) which is pretrained on full input tensor \( X \)
- Two loss functions:
  - 2-norm regression loss
    \[ \ell_2(F(X_1), X_2) = \frac{1}{h,w} \sum_{h,w} \| X_{2h,w} - F(X_1)_{h,w} \|^2 \]
  - Quantized cross-entropy loss
    \[ \ell_{cd}(F(X_1), X_2) = -\sum_{h,w} \sum_{q} \mathcal{H}(X_2)_{h,w,q} \log(F(X_1)_{h,w,q}) \]
    \[ \mathcal{H}(X_2), F(X_1) \in \Delta^{H \times W \times Q} \]

\( Q \) is the number of elements in the quantized output space.
Approach

• Train on 1.3M ImageNet dataset [36] (without labels)
• Use Lab color space. L(1D): lightness channel ab(2D): color channel
• Use AlexNet architecture (five conv layers + three dense layers)

Experiments – RGB images

• Representation learning benchmarks:
  • ImageNet classification (1000-way) (same dataset)
    Training multinomial logistic regression classifiers on top of each conv layer.
  • Largescale Places dataset classification
  • PASCAL dataset
    • Classification: 20 binary classification decisions
    • Detection: localized bounding box around any objects
    • Segmentation: pixel-wise labeling of the object class among 20
Experiments – RGB images

Comparison to Previous Unsupervised Methods

Experiments – RGB images

Ablation Studies on ImageNet classification
Experiments – RGB-D data

<table>
<thead>
<tr>
<th>Method</th>
<th>Data</th>
<th>Label</th>
<th>RGB</th>
<th>D</th>
<th>RGB-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gupta et al. [17]</td>
<td>1M ImNet [36]</td>
<td>✓</td>
<td>27.8</td>
<td>41.7</td>
<td>47.1</td>
</tr>
<tr>
<td>Gupta et al. [16]</td>
<td>1M ImNet [36]</td>
<td>✓</td>
<td>27.8</td>
<td>34.2</td>
<td>44.4</td>
</tr>
<tr>
<td>Gaussian</td>
<td>None</td>
<td>–</td>
<td>–</td>
<td>28.1</td>
<td>–</td>
</tr>
<tr>
<td>Krähenbühl et al. [24]</td>
<td>20 NYU-D [38]</td>
<td>12.5</td>
<td>32.2</td>
<td>34.5</td>
<td></td>
</tr>
<tr>
<td>Split-Brain Autoencoder</td>
<td>10k NYU-D [38]</td>
<td>18.9</td>
<td>33.2</td>
<td>38.1</td>
<td></td>
</tr>
</tbody>
</table>

Split-Brain Autoencoder Results on RGB-D images

Conclusion

The method
(i) does not require a representational bottleneck for training,
(ii) uses input dropout to help force abstraction in the representation,
(iii) is pre-trained on the full input data.