Zero-Shot Detection via Vision and Language Knowledge Distillation

ViLD: Vision and Language Knowledge Distillation

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, Yin Cui
Google Research

MSR V+L Summer Talk Series, July 28, 2021
Zero-shot open vocabulary detection

- toy elephant: 0.51
- blue toy: 0.74
- toy: 0.77
- toy crocodile: 0.20
- green toy: 0.48
- toy: 0.56
- toy duck: 0.86
- yellow toy: 0.70
- toy: 0.89

- : Novel categories
- : Base categories
Motivation
## Dataset collection for large vocabulary detection

<table>
<thead>
<tr>
<th>Dataset</th>
<th># images</th>
<th># boxes</th>
<th># categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pascal VOC</td>
<td>11.5k</td>
<td>27k</td>
<td>20</td>
</tr>
<tr>
<td>COCO</td>
<td>159k</td>
<td>896k</td>
<td>80</td>
</tr>
<tr>
<td>Objects 365</td>
<td>1800k</td>
<td>29,000k</td>
<td>365</td>
</tr>
<tr>
<td>LVIS v1.0</td>
<td>159k</td>
<td>1,514k</td>
<td>1203</td>
</tr>
</tbody>
</table>
Long-tailed distribution

Zipf’s Law

- Natural object categories follow a long-tailed distribution.
- Exponentially more data is needed for rare categories.
- Expensive to scale up dataset vocabularies.
- Alternatives?

Number of instances per category in LVIS dataset
Open vocabulary detection can be a new direction for large vocabulary detection
Recent zero-shot classification models

- CLIP (OpenAI)
- 400M image-text pairs

1. **Contrastive pre-training**

---

https://openai.com/blog/clip/

Recent zero-shot classification models

- CLIP (OpenAI)
- 76.2% Top-1 Acc on ImageNet
Recent zero-shot classification models

- ALIGN (Google)

- 1.8B noisier image-text pairs

- Comparable performance with CLIP
Borrowing the knowledge from zero-shot classification model for zero-shot detection
Method
Settings for zero-shot detection

- **Base categories** the model can be trained on.

- **Novel categories** that are never seen during training.

- **Goal**: achieve good performance on novel categories while maintaining performance on base categories.

Note: Our method is zero-shot w.r.t. the detection dataset. Our method does not learn from detection annotations of novel categories.

However, similar concepts of novel categories could be seen in the pre-trained zero-shot classification models (e.g., CLIP)
Object proposals for novel categories

- Two-stage object detector (e.g., Mask R-CNN).
- *Class-agnostic* bbox regression and mask prediction.

He et al., 2017. Mask R-CNN.
A straightforward approach: Zero-shot detection with cropped regions

- Ensemble 1x and 1.5x crops (to include context).
- Similar to R-CNN.
- Slow!
- Not utilizing annotations from base categories.
Leveraging pre-trained zero-shot classification model

- CLIP

1. Contrastive pre-training

- Text Encoder
  - pepper the aussie pup

- Image Encoder
  - Image $I_i$ ($I_1, I_2, I_3, ..., I_N$)
  - Text $T_j$ ($T_1, T_2, T_3, ..., T_N$)

References:

https://openai.com/blog/clip/
ViLD-text

- **Text prompts**: e.g., “a photo of a {category} in the scene”.
- **Category text embeddings**: feed the text prompts into the pre-trained text encoder.
  - Ensemble 63 text prompts, with synonyms if available.
- **Learnable “background” embedding**: for proposals do not match any labeled categories.
- **Classify with text embeddings**:
  \[
  L_{CE}(\text{softmax}(1/T \cdot \cosine\_similarity(region\ embedding, text/background\ embeddings)))
  \]
ViLD-image

Vanilla detector
Cross entropy loss

Classifier
N region embeddings
Head
N proposals

ViLD
cross entropy loss

Back ground
Text Embeddings
N region embeddings
L2 Normalization
Projection Layer
Head
N proposals

ViLD-text

Knowledge Distillation

M region embeddings
Splitting

L1 loss

Pre-trained Image Encoder

M image embeddings
Cropping & Resizing

M pre-computed proposals for distillation

Cropped regions
ViLD overview

Training

Cropped Regions

Pre-trained Image Encoder

Knowledge Distillation

$L_1$ loss

Cross entropy loss

Base Categories
- stop sign
- car
- dice
- lapepian herder

Novel Categories

A photo of a [category] in the scene

Pre-trained Text Encoder

Inference

Backbone + RPN

RoIAlign

Conv

$B_1$, $B_2$, $B_3$, $B_n$

$N_1$, $N_2$, $N_3$, $N_k$

$R_1$, $R_2$, $R_3$, $R_n$

image embeddings

region embeddings

text embeddings

dice

lapepian herder
Model ensembling to mitigate conflicting objectives

- 1) ViLD-text + CLIP:
  - CLIP is the teacher model for ViLD-image.
  - Slow!

- 2) ViLD-ensemble: Two separate heads for ViLD-text and ViLD-image objectives, respectively.
  - Weighted average: For base categories, weigh the predictions of ViLD-text more; vice versa for novel categories.
Results
Benchmark settings

- Main dataset: LVIS v1.0 (1203 categories)
  - Frequent (f: 405 classes, 100-1977 images per class) and common (c: 461 classes, 10-100 images per class) categories as base categories
  - Rare (r: 337 classes, <10 images per class) categories as novel categories
- Metrics: Average Precision (AP), $\text{AP}_r$, $\text{AP}_c$, $\text{AP}_f$
Object proposals for novel categories

RPN’s Average Recall (AR) for novel categories

<table>
<thead>
<tr>
<th>Supervision</th>
<th>$AR_r@100$</th>
<th>$AR_r@300$</th>
<th>$AR_r@1000$</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>39.3</td>
<td>48.3</td>
<td>55.6</td>
</tr>
<tr>
<td>base + novel</td>
<td>41.1</td>
<td>50.9</td>
<td>57.0</td>
</tr>
</tbody>
</table>

- RPN trained on base categories generalizes to novel categories, yielding higher scores for unseen categories compared with background.
Classifying proposals with CLIP

<table>
<thead>
<tr>
<th>Method</th>
<th>$AP_r$</th>
<th>$AP_c$</th>
<th>$AP_f$</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised (base class only)</td>
<td>0.0</td>
<td>22.6</td>
<td>32.4</td>
<td>22.5</td>
</tr>
<tr>
<td>CLIP on cropped regions</td>
<td>13.0</td>
<td>10.6</td>
<td>6.0</td>
<td>9.2</td>
</tr>
<tr>
<td>Supervised (base + novel)</td>
<td>4.1</td>
<td>23.5</td>
<td>33.2</td>
<td>23.9</td>
</tr>
<tr>
<td>Supervised-RFS (base + novel)</td>
<td>12.3</td>
<td>24.3</td>
<td>32.4</td>
<td>25.4</td>
</tr>
</tbody>
</table>

- Supervised-RFS: a better supervised baseline with Repeat Factor Sampling to upsample rare classes.
- $AP_r$ is on par with supervised learning approaches.
- Overall performance is behind.
A strong baseline (large-scale jittering + longer training)

- Formally introduced in our Copy-Paste data augmentation paper [1] and recently adopted by Detectron2
- Recent FB AI Blog: *Advancing computer vision research with new Detectron2 Mask R-CNN baselines*

Recent work in the field, such as [Simple Copy-Paste Data Augmentation](https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md), has shown substantial improvements in accuracy (measured by average precision, or AP) for two core tasks, creating a bounding box around an object and drawing a detailed mask over different objects. The paper's highest-reported Mask R-CNN ResNet-50-FPN baseline is 47.2 Box AP and 41.8 Mask AP, which exceeds Detectron2’s highest reported baseline of 41.0 Box AP and 37.2 Mask AP. This difference is significant because most research papers publish improvements in the order of 1 percent to 3 percent.

New baselines using Large-Scale Jitter and Longer Training Schedule

The following baselines of COCO Instance Segmentation with Mask R-CNN are generated using a longer training schedule and large-scale jitter as described in Google’s *Simple Copy-Paste Data Augmentation* paper. These models are trained from scratch using random initialization. These baselines exceed the previous Mask R-CNN baselines.

In the following table, one epoch consists of training on 118000 COCO images.

<table>
<thead>
<tr>
<th>Name</th>
<th>epochs</th>
<th>train time (s/ep)</th>
<th>inference time (s/ep)</th>
<th>box AP</th>
<th>mask AP</th>
<th>model id</th>
<th>download</th>
</tr>
</thead>
<tbody>
<tr>
<td>R50-FPN</td>
<td>100</td>
<td>0.0376</td>
<td>0.069</td>
<td>44.6</td>
<td>40.3</td>
<td>42047764</td>
<td>model</td>
</tr>
<tr>
<td>R50-FPN</td>
<td>200</td>
<td>0.0376</td>
<td>0.069</td>
<td>46.3</td>
<td>41.7</td>
<td>42047738</td>
<td>model</td>
</tr>
<tr>
<td>R50-FPN</td>
<td>400</td>
<td>0.0376</td>
<td>0.069</td>
<td>47.4</td>
<td>42.6</td>
<td>42019571</td>
<td>model</td>
</tr>
<tr>
<td>R101-FPN</td>
<td>100</td>
<td>0.518</td>
<td>0.073</td>
<td>49.4</td>
<td>41.6</td>
<td>42025812</td>
<td>model</td>
</tr>
<tr>
<td>R101-FPN</td>
<td>200</td>
<td>0.518</td>
<td>0.073</td>
<td>49.0</td>
<td>43.1</td>
<td>42131867</td>
<td>model</td>
</tr>
<tr>
<td>R101-FPN</td>
<td>400</td>
<td>0.518</td>
<td>0.073</td>
<td>48.9</td>
<td>43.7</td>
<td>42073830</td>
<td>model</td>
</tr>
</tbody>
</table>

[1] Ghiasi et al., 2021. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation

https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md
ViLD-text

Backbone: Mask R-CNN R50-FPN

<table>
<thead>
<tr>
<th>Method</th>
<th>$\text{AP}_r$</th>
<th>$\text{AP}_c$</th>
<th>$\text{AP}_f$</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIP on cropped regions</td>
<td>13.0</td>
<td>10.6</td>
<td>6.0</td>
<td>9.2</td>
</tr>
<tr>
<td>GloVe baseline</td>
<td>3.0</td>
<td>20.1</td>
<td>30.4</td>
<td>21.2</td>
</tr>
<tr>
<td><strong>ViLD-text</strong></td>
<td><strong>10.1</strong></td>
<td><strong>23.9</strong></td>
<td><strong>32.5</strong></td>
<td><strong>24.9</strong></td>
</tr>
<tr>
<td>Supervised-RFS (base + novel)</td>
<td><strong>12.3</strong></td>
<td><strong>24.3</strong></td>
<td><strong>32.4</strong></td>
<td><strong>25.4</strong></td>
</tr>
</tbody>
</table>

- **Outperforms GloVe baseline:**
  - text embeddings jointly trained with visual data.
- **Outperforms CLIP on cropped regions:**
  - trained with annotations from base categories.
ViLD-image

Backbone: Mask R-CNN R50-FPN

<table>
<thead>
<tr>
<th>Method</th>
<th>AP_r</th>
<th>AP_c</th>
<th>AP_f</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIP on cropped regions</td>
<td>13.0</td>
<td>10.6</td>
<td>6.0</td>
<td>9.2</td>
</tr>
<tr>
<td>GloVe baseline</td>
<td>3.0</td>
<td>20.1</td>
<td>30.4</td>
<td>21.2</td>
</tr>
<tr>
<td>ViLD-text</td>
<td>10.1</td>
<td>23.9</td>
<td>32.5</td>
<td>24.9</td>
</tr>
<tr>
<td><strong>ViLD-image</strong></td>
<td><strong>9.6</strong></td>
<td><strong>8.5</strong></td>
<td><strong>7.8</strong></td>
<td><strong>8.4</strong></td>
</tr>
<tr>
<td>Supervised-RFS (base + novel)</td>
<td>12.3</td>
<td>24.3</td>
<td>32.4</td>
<td>25.4</td>
</tr>
</tbody>
</table>

- Only trained with L1 distillation loss, no cross entropy loss.
- Ideally, similar performance as CLIP on cropped regions.
- A small performance gap.
ViLD

Backbone: Mask R-CNN R50-FPN

<table>
<thead>
<tr>
<th>Method</th>
<th>AP$_r$</th>
<th>AP$_c$</th>
<th>AP$_f$</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIP on cropped regions</td>
<td>13.0</td>
<td>10.6</td>
<td>6.0</td>
<td>9.2</td>
</tr>
<tr>
<td>GloVe baseline</td>
<td>3.0</td>
<td>20.1</td>
<td>30.4</td>
<td>21.2</td>
</tr>
<tr>
<td>ViLD-text</td>
<td>10.1</td>
<td>23.9</td>
<td>32.5</td>
<td>24.9</td>
</tr>
<tr>
<td>ViLD-image</td>
<td>9.6</td>
<td>8.5</td>
<td>7.8</td>
<td>8.4</td>
</tr>
<tr>
<td>ViLD ($w = 0.5$)</td>
<td>16.1</td>
<td>20.0</td>
<td>28.3</td>
<td>22.5</td>
</tr>
<tr>
<td>Supervised-RFS (base + novel)</td>
<td>12.3</td>
<td>24.3</td>
<td>32.4</td>
<td>25.4</td>
</tr>
</tbody>
</table>

- Outperforms supervised counterpart on novel categories!
- AP$_r$ improved over ViLD-text or ViLD-image.
ViLD

Hyperparameter sweep

<table>
<thead>
<tr>
<th>Distill loss</th>
<th>Distill weight $w$</th>
<th>$AP_r$</th>
<th>$AP_c$</th>
<th>$AP_f$</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>No distill</td>
<td>0.0</td>
<td>10.4</td>
<td>22.9</td>
<td>31.3</td>
<td>24.0</td>
</tr>
<tr>
<td>$\mathcal{L}_2$ loss</td>
<td>0.5</td>
<td>13.7</td>
<td>21.7</td>
<td>31.2</td>
<td>24.0</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>12.4</td>
<td>22.7</td>
<td>31.4</td>
<td>24.3</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>13.4</td>
<td>22.0</td>
<td>30.9</td>
<td>24.0</td>
</tr>
<tr>
<td>$\mathcal{L}_1$ loss</td>
<td>0.05</td>
<td>12.9</td>
<td>22.4</td>
<td>31.7</td>
<td>24.4</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>14.0</td>
<td>20.9</td>
<td>31.2</td>
<td>23.8</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>16.3</td>
<td>19.2</td>
<td>27.3</td>
<td>21.9</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>17.3</td>
<td>18.2</td>
<td>25.1</td>
<td>20.7</td>
</tr>
</tbody>
</table>

- L1 loss is better than L2 loss.
- Trend: as $w$ increases, $AP_r \uparrow$, $AP_f$ and $AP_c \downarrow \rightarrow$

A competition between ViLD-text and ViLD-image.

- We later mitigate the competition by ensembling.
# Model ensembling

## Backbone: Mask R-CNN R50-FPN

<table>
<thead>
<tr>
<th>Method</th>
<th>AP&lt;sub&gt;r&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;c&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;f&lt;/sub&gt;</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIP on cropped regions</td>
<td>13.0</td>
<td>10.6</td>
<td>6.0</td>
<td>9.2</td>
</tr>
<tr>
<td>GloVe baseline</td>
<td>3.0</td>
<td>20.1</td>
<td>30.4</td>
<td>21.2</td>
</tr>
<tr>
<td>ViLD-text</td>
<td>10.1</td>
<td>23.9</td>
<td>32.5</td>
<td>24.9</td>
</tr>
<tr>
<td>ViLD-image</td>
<td>9.6</td>
<td>8.5</td>
<td>7.8</td>
<td>8.4</td>
</tr>
<tr>
<td>ViLD (w = 0.5)</td>
<td>16.1</td>
<td>20.0</td>
<td>28.3</td>
<td>22.5</td>
</tr>
<tr>
<td><strong>ViLD-ensemble (w = 0.5)</strong></td>
<td><strong>16.6</strong></td>
<td><strong>24.6</strong></td>
<td><strong>30.3</strong></td>
<td><strong>25.5</strong></td>
</tr>
<tr>
<td><strong>ViLD-text + CLIP</strong></td>
<td><strong>22.6</strong></td>
<td><strong>24.8</strong></td>
<td><strong>29.2</strong></td>
<td><strong>26.1</strong></td>
</tr>
<tr>
<td>Supervised-RFS (base + novel)</td>
<td>12.3</td>
<td>24.3</td>
<td>32.4</td>
<td>25.4</td>
</tr>
</tbody>
</table>

- ViLD-text + CLIP attains the best AP<sub>r</sub> and good overall AP.
  - 630x slower.
- ViLD-ensemble improves AP<sub>c</sub> and AP<sub>f</sub> over ViLD.
Results with Mask R-CNN R152-FPN backbone

<table>
<thead>
<tr>
<th>Method</th>
<th>(\text{AP}_r)</th>
<th>(\text{AP}_c)</th>
<th>(\text{AP}_f)</th>
<th>(\text{AP})</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViLD-text</td>
<td>11.7</td>
<td>25.8</td>
<td>34.4</td>
<td>26.7</td>
</tr>
<tr>
<td>ViLD-image</td>
<td>10.8</td>
<td>10.0</td>
<td>8.7</td>
<td>9.6</td>
</tr>
<tr>
<td>ViLD ((w = 1.0))</td>
<td><strong>18.7</strong></td>
<td>21.1</td>
<td>28.4</td>
<td>23.6</td>
</tr>
<tr>
<td>ViLD-ensemble ((w = 2.0))</td>
<td><strong>18.7</strong></td>
<td>24.9</td>
<td>30.6</td>
<td>26.0</td>
</tr>
<tr>
<td>Supervised-RFS (base + novel)</td>
<td>14.4</td>
<td>26.8</td>
<td>34.2</td>
<td>27.6</td>
</tr>
</tbody>
</table>

- \(\text{AP}_r\) further improves with stronger backbones.
- Same trend as R50.
Transfer to other detection datasets

- Replace with category text embeddings of a new dataset.
- **A finetuning-free transfer!**
- Small gaps compared with finetuning (start from ViLD, finetune the linear classifier).

<table>
<thead>
<tr>
<th>Method</th>
<th>PASCAL VOC† AP&lt;sub&gt;50&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;75&lt;/sub&gt;</th>
<th>COCO AP&lt;sub&gt;50&lt;/sub&gt; AP&lt;sub&gt;75&lt;/sub&gt; AP&lt;sub&gt;s&lt;/sub&gt; AP&lt;sub&gt;m&lt;/sub&gt; AP&lt;sub&gt;t&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;50&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;75&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;s&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;m&lt;/sub&gt;</th>
<th>AP&lt;sub&gt;t&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViLD-text</td>
<td>40.5</td>
<td>31.6</td>
<td>28.8 43.4 31.4 11.8 35.5 52.0</td>
<td>10.4</td>
<td>15.8</td>
<td>11.1</td>
<td>4.5</td>
<td>11.4</td>
</tr>
<tr>
<td>ViLD</td>
<td>72.2</td>
<td>56.7</td>
<td>36.6 55.6 39.8 20.7 39.2 52.6</td>
<td>11.8</td>
<td>18.2</td>
<td>12.6</td>
<td>5.5</td>
<td>13.5</td>
</tr>
<tr>
<td>Finetuning</td>
<td>78.9</td>
<td>60.3</td>
<td>39.1 59.8 42.4 21.0 41.7 55.0</td>
<td>15.2</td>
<td>23.9</td>
<td>16.2</td>
<td>7.3</td>
<td>17.2</td>
</tr>
<tr>
<td>Supervised</td>
<td>78.5</td>
<td>49.0</td>
<td>46.5 67.6 50.9 27.1 67.6 77.7</td>
<td>25.6</td>
<td>38.6</td>
<td>28.0</td>
<td>16.0</td>
<td>28.1</td>
</tr>
</tbody>
</table>
Qualitative examples
On-the-fly interactive detection

- After detecting pre-defined categories, use *on-the-fly free-form* text embeddings to recognize more details.
Systematic expansion of dataset vocabulary

- Dataset vocabulary: $v = \{v_1, \ldots, v_p\}$.
- Attributes set: $a = \{a_1, \ldots, a_q\}$.
- Given region embedding $e_r$.
  
  \[
  \Pr(v, a | e_r) = \Pr(v_i | e_r) \times \Pr(a_j | e_r).
  \]
- Expand $p$ vocabularies into $p \times q$ vocabularies.
Systematic expansion of dataset vocabulary

- Detect fruit with color attributes (expand LVIS vocabulary with 11 colors).

Original dataset vocabulary

Expanded with color attributes
Systematic expansion of dataset vocabulary

- 200 Fine-grained bird species from CUB-200-2011 (expanded from LVIS vocabulary).

(a) Successful cases

(b) Failure case
Failure cases

- **red**: groundtruth of failed detections for novel objects

(a) Missed

(b) Misclassified
Compare w/ existing zero-shot detection methods on COCO

- An issue of zero-shot detention on COCO: people are using different splits of COCO dataset
- ViLD-text alone outperforms the most recent SOTA under the same setting (65/15 split, IoU=0.5), especially Recall

<table>
<thead>
<tr>
<th>Method</th>
<th>Unseen (mAP/Recall@100)</th>
<th>Seen (mAP/Recall@100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rahman <em>et al.</em></td>
<td>12.40 / 37.16</td>
<td>34.07 / 36.38</td>
</tr>
<tr>
<td>ViLD-text</td>
<td>14.71 / 47.69</td>
<td>43.62 / 85.55</td>
</tr>
</tbody>
</table>
An improved version with ALIGN

- ViLD-ensemble on LVIS v1.0 with a pre-trained ALIGN model

<table>
<thead>
<tr>
<th>Method</th>
<th>Image Model for Distillation</th>
<th>Detector Backbone</th>
<th>APr (novel)</th>
<th>APc (base)</th>
<th>APf (base)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViLD-ensemble</td>
<td>CLIP (ViT-B/32)</td>
<td>ResNet-152</td>
<td>18.7</td>
<td>24.9</td>
<td>30.6</td>
</tr>
<tr>
<td>ViLD-ensemble</td>
<td>ALIGN (EfficientNet-B7)</td>
<td>EfficientNet-B7</td>
<td>26.3</td>
<td>27.2</td>
<td>32.9</td>
</tr>
<tr>
<td>Supervised Baseline</td>
<td>-</td>
<td>EfficientNet-B7</td>
<td>15.4 (-10.9)</td>
<td>27.8 (+0.6)</td>
<td>34.3 (+1.4)</td>
</tr>
</tbody>
</table>
An improved version with ALIGN - qualitative example

Input texts

category_indices:
1: {'id': 1, 'name': 'black flipflop'},
2: {'id': 2, 'name': 'white flipflop'},
3: {'id': 3, 'name': 'street sign'},
4: {'id': 4, 'name': 'bracelet'},
5: {'id': 5, 'name': 'necklace'},
6: {'id': 6, 'name': 'shorts'},
7: {'id': 7, 'name': 'flouery top'},
8: {'id': 8, 'name': 'blue dress'},
9: {'id': 9, 'name': 'orange shirt'},
10: {'id': 10, 'name': 'purple dress'},
11: {'id': 11, 'name': 'yellow tshirt'},
12: {'id': 12, 'name': 'green umbrella'},
13: {'id': 13, 'name': 'pink striped umbrella'},
14: {'id': 14, 'name': 'plain white umbrella'},
15: {'id': 15, 'name': 'plain pink umbrella'},
16: {'id': 16, 'name': 'blue patterned umbrella'},
17: {'id': 17, 'name': 'koala'}}
Conclusion

- ViLD: an open-vocabulary detection method by distilling knowledge from a zero-shot image classification model.
- Achieves 18.7 AP\textsubscript{r} on LVIS (R152-FPN), surpassing supervised counterpart at the same inference speed.
- Append new classes without re-training of the detector.
- Transfers to other datasets without fine-tuning.
- Enables free-form text detection.
- An alternative for detecting long-tailed classes, rather than scaling up detection datasets by collecting exponentially more images to cover long-tail classes.
Thank You