Knowledge transfer and human-machine collaboration for object detection and segmentation

June 2018

Vittorio Ferrari
Google Research

CVPR 2018 WebVision workshop
Manual annotation is expensive

Training modern models requires:

26s (ImageNet)
80s (COCO)
1000s (COCO-Stuff)

Su et al., Crowdsourcing annotations for visual object detection, AAAI 2012
Deng et al., Scalable multi-label annotation, CHI 2014
Lin et al., Microsoft COCO: common objects in context, ECCV 2014
Caesar et al., COCO-Stuff: Things and Stuff classes in context, CVPR 2018
Papadopoulos et al., Extreme Clicking for efficient object annotation, ICCV 2017: box in 7s!
Fully supervised learning

Annotation to the same degree as outputs on test images
Weakly supervised learning

Annotation to a lower degree than outputs on test images

Motorbike

Object detection model

Panoptic segmentation model

Person, book, wall, light, ...

Sky, horse, train, ...

Annotation to a lower degree than outputs on test images
Weakly supervised learning

Core issue: need for auto-annotation
Human-machine collaboration

Object detection model

Panoptic segmentation model

Human intervenes during machine process
Weak Supervision for bounding-boxes

• basic: image-level labels
  [Nguyen ICCV 09, Deselaers ECCV 10, Siva ICCV 11, 
   Song ICML 14, Cinbis CVPR 14, Wang TIP 15, 
   Bilen CVPR 16, Kantorov ECCV 16, Dong ACMMM 17]

  + point click on object
    [Mettes ECCV 16, Papadopoulos CVPR 17]

  + video
    [Prest CVPR 12, Tang CVPR 13, Joulin ECCV 14, 
     Kuznetsova CVPR 15, Liang ICCV 15, 
     Liang ICCV 15, Kalogeiton PAMI 15]

  + knowledge transfer from classes with boxes
    [Salakhutdinov CVPR 11, Aytar ICCV 11, Guillaumin CVPR 12, 
     Vezhnevets CVPR 14, Hoffman NIPS 14, Rochan CVPR 15, 
     Tang CVPR 16, Redmon CVPR 17]

  + eye-tracks
    [Papadopulous ECCV 14, Mathe arXiv 14, 
     Karthikeyan CVPR 15]
Weak Supervision for object segments

- **basic: image-level labels**

- **point click on object**
  [Wang CVIU 14, Bell CVPR 15, Bearman ECCV 16, Jain AAAI 16]

- **scribbles**
  [Xu CVPR 15, Lin CVPR 16]

- **object bounding-boxes**
  [Dai ICCV 15, Khoreva CVPR 17]

- **extreme points on object**
  [Papadopoulos ICCV 17, Maninis CVPR 18]

- **transfer from other pre-segmented classes**
  [Kuettel ECCV 12, Rubinstein ECCV 12]

- **video**
  [Tokmakov ECCV 16]
Human-Machine collaboration

- classic active learning (ask label of samples)
  for box: [Vijayanarasimhan IJCV 14, Yao CVPR 12]
  for segmentations [Vezhnevets CVPR 12, Jain CVPR 16]

- box verification series
  [Papadopoulos CVPR 16]

- select which annotation micro-task to ask for
  [Vijayanarasimhan CVPR 09, Jain ICCV 13, Russakovsky CVPR 15]

- interactive object segmentation
  [Boykov ICCV 01, Rother SIGGRAPH 04, Wang ICCV 05, Liew ICCV 17, Xu BMVC 17, Castrejon CVPR 17, Li CVPR 18]

- fine-grained classification by asking attributes
  [Branson ECCV 10, Biswas CVPR 13, Wah CVPR 14]
This talk

- Revisiting knowledge transfer for training object class detectors
  Uijlings, Popov, Ferrari
  CVPR 2018

- Learning intelligent dialogs for bounding box annotation
  Konyushkova, Uijlings, Lampert, Ferrari
  CVPR 2018

- Fluid Annotation: human-machine collaboration for full image annotation
  Andriluka, Uijlings, Ferrari,
  arXiv June 2018
Problem setting: knowledge transfer

**Source** train set (bounding boxes)

- bicycle
- cat

**Target** train set (image-level labels)

- motorbike

**Intermediate goal:** Localize target class in train set

- motorbike

**Goal:** detect target class in test set

- Target test set (no labels)

- [Guillaumin CVPR 12, Hoffman NIPS 14, Rochan CVPR 15, Tang CVPR 16, Redmon CVPR 17]
Why relevant?

- Image-level labels are cheaper to obtain (2s)

- But weakly supervised methods lead to lower quality detectors (~60% the mAP of full supervision)

- There are large datasets with bounding boxes

[Deselaers ECCV 10, Bilen CVPR 15, Blaschko NIPS 10, Cinbis CVPR 14, Nguyen ICCV 09, Pandey ICCV 11, Russakovsky ECCV 12, Shi PAMI15, Shi BMVC 12, Siva CVPR 13, Siva ICCV 11, Siva ECCV 12, Song NIPS 14, Song ICML 14, Wang TIP 15, Bilen CVPR 16, Dong ACMMM 17]
A typical WSOL framework

Positive images  Negative images  Multiple instance learning (MIL)

images = bags
windows = instances

Object proposals
[Alexe CVPR 10, Dollar ECCV 14, van de Sande ICCV 11, …]

Goals:
• find true positive instances
• train window classifier

[Blaschko NIPS 10, Cinbis PAMI 16, Deselaers ECCV 10, Nguyen ICCV 09, Russakovsky ECCV 12, Shi PAMI 15, Shi BMVC 12, Siva ICCV 11 & ECCV 12, Song NIPS 14 & ICML 14, Bilen BMVC 14, Sangineto PAMI 18, Li BMVC 17]
MIL-style WSOL

Re-training object detectors

Initialization: full images
[Cinbis CVPR 14, Nguyen ICCV 09, Russakovsky ECCV 12, Pandey ICCV 11]

Re-localizing objects

Objectness and multi-folding
[Deselaers ECCV 10] [Cinbis CVPR 14]

Re-localization: pick proposal with highest appearance score

[Blaschko NIPS 10, Cinbis PAMI 16, Deselaers ECCV 10, Nguyen ICCV 09, Russakovsky ECCV 12, Shi PAMI 15, Shi BMVC 12, Siva ICCV 11 & ECCV 12, Song NIPS 14 & ICML 14, Bilen BMVC 14, Sangineto PAMI 18, Li BMVC 17]
MIL-style WSOL

\[ f(b) = \lambda A(b, I) + (1 - \lambda)O(b, I) \]

- Re-training detector
- Re-localizing objects
- Appearance model
- Object proposals

[Alexe CVPR 10]
[Zitnick ECCV 14]
MIL-style WSOL

Re-training detector

Re-localizing objects

\[ f(b) = \lambda A(b, I) + (1 - \lambda) O(b, I) \]

Object proposals

[Alexe CVPR 10]
[Zitnick ECCV 14]

[Deselaers ECCV 10, Prest CVPR 12, Shapovalova ECCV 12, Shi BMVC 12, Siva ICCV 11, Cinbis CVPR 14, Bilen CVPR 16, Tang CVPR 14, Wang ECCV 14]
MIL-style WSOL

Re-training detector

Object proposals

[Alexe CVPR 10]
[Zitnick ECCV 14]

Re-localizing objects

\[ f(b) = \lambda A(b, I) + (1 - \lambda)O(b, I) \]

Manually engineered measure

[Deselaers ECCV 10, Prest CVPR 12, Shapovalova ECCV 12, Shi BMVC 12, Siva ICCV 11, Cinbis CVPR 14, Bilen CVPR 16, Tang CVPR 14, Wang ECCV 14]
Problem setting: knowledge transfer

**Source** train set (bounding boxes)

- bicycle
- cat

**Target** train set (image-level labels)

- motorbike
- no motorbike

**Target test set** (no labels)

**Goal**: detect target class in test set

[Guillaumin CVPR 12, Hoffman NIPS 14, Rochan CVPR 15, Tang CVPR 16, Redmon CVPR 17]
Knowledge Transfer

Source train set

- animal
- giraffe
- bicycle
- quad
- vehicle
- entity
- animal

Target train set

- entity: 0.5
- giraffe: 0.2
- bicycle: 0.0
- quad: 0.0
- vehicle: 0.0

Train Multibox SSD on all classes in hierarchy

Related:
- Guillaumin CVPR 12
- Hoffman NIPS 14
Re-localizing objects

\[ f(b) = \lambda A(b, I) + (1 - \lambda) K(b, I) \]

MIL + Knowledge Transfer
MIL + Knowledge Transfer

Re-localizing objects

\[ f(b) = \lambda A(b, I) + (1 - \lambda) K(b, I) \]

Source detectors
- entity
- vehicle
- bicycle

Learned measure
MIL + Knowledge Transfer

Re-localizing objects

\[ f(b) = \lambda A(b, I) + (1 - \lambda) K(b, I) \]

Learned measure

Source detectors
- entity
- vehicle
- bicycle

Re-training detector
Dataset: ILSVRC 2013

Setup following LSDA [Hoffman NIPS 14]

**source training set**

- val1 augmented to 1000 boxes per class [Girshick CVPR 14]
- classes 1-100
- bounding box annotations
- for each knowledge transfer function, optimize $\lambda$ on 80/20 class split (and all other hyper-parameters ;)

**target training set**

- val1 augmented to 1000 boxes per class [Girshick CVPR 14]
- classes 101-200
- Image-labels only

**target test set**

- Complete val2 set
- Accuracy measured on class 101-200 only
Results: Qualitative localizations on target train

- rubber eraser
- watercraft
- violin
- tie

Methods:
- EdgeBox + objectness baseline
- Knowledge Transfer (class-generic)
CorLoc@0.5 on target training set

- EdgeBox (no score)
- EdgeBox + Objectness
- No score
- Closest source
- Closest ancestor
- Class generic
Delivers detectors reaching 80% of mAP of fully supervision!
Semantic similarity vs improvement

✓ improvement uncorrelated with semantic similarity
Comparison to state-of-the-art

- CorLoc @ IoU > 0.5 on target training set
  - LSDA [Hoffman NIPS 14]
  - Class generic

- mAP @ IoU > 0.5 on target test set
  - LSDA [Hoffman NIPS 14]
  - Tang CVPR 16
  - Class generic

Faster-RCNN + Inception ResNet

AlexNet backbone
Comparison to YOLOv2 [Redmon ICCV 17]

Source training set

Target training set

Target test set

ILSVRC detection validation set:
156 Non-COCO classes
## Generalization across datasets: CorLoc @ 0.5

<table>
<thead>
<tr>
<th>Source training set</th>
<th>Target training set</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IMAGENET</strong></td>
<td>Augmented val 1, class 101-200</td>
</tr>
<tr>
<td>Augmented val 1, class 1-100</td>
<td>74.2</td>
</tr>
<tr>
<td>100 classes, 65k images, 81k boxes</td>
<td>34.5</td>
</tr>
<tr>
<td><strong>COCO</strong></td>
<td>2014 train</td>
</tr>
<tr>
<td>2014 train</td>
<td>67.7</td>
</tr>
<tr>
<td>80 classes, 83k images, 605k boxes</td>
<td>-</td>
</tr>
<tr>
<td><strong>VOC 2007 trainval</strong></td>
<td>59.5</td>
</tr>
<tr>
<td>20 classes, 5k images, 13k boxes</td>
<td>26.2</td>
</tr>
<tr>
<td><strong>EdgeBox baseline</strong></td>
<td>50.5</td>
</tr>
<tr>
<td></td>
<td>20.6</td>
</tr>
<tr>
<td></td>
<td>32.4</td>
</tr>
</tbody>
</table>
Conclusions

- Knowledge Transfer substantially improves Weakly Supervised Object Detection

- Delivers detectors performing at 80% of the mAP of their fully supervised counterparts

- Generalize across a wide range of source-target dataset pairs

- Simple modification to standard MIL pipelines
This talk

- Revisiting knowledge transfer for training object class detectors
  Uijlings, Popov, Ferrari
  CVPR 2018

- Learning intelligent dialogs for bounding box annotation
  Konyushkova, Uijlings, Lampert, Ferrari
  CVPR 2018

- Fluid Annotation: human-machine collaboration for full image annotation
  Andriluka, Uijlings, Ferrari,
  arXiv June 2018
Manual box drawing

ImageNet annotation timings: [Su AAAI 2012]
Crowdsourcing annotations for visual object detection. AAAI Human Computation Workshop 2012
Box verification series

D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari.
We don’t need no bounding-boxes: Training object class detectors using only human verification. CVPR 2016
Success, failure and motivation

cat

potted plant

Depends on image difficulty, detector strength, desired box quality
Intelligent Annotation Dialog (IAD)

image

Detector

class: boat

box proposals

agent

Verify

Draw

??

no

no

yes

yes

END
Agent 1: Model-based

Given:
- time for drawing \( t_{\text{draw}} \)
- time for verification \( t_{\text{verify}} \)
- probability of acceptance of box \( i \) \( p_i \)

we can estimate the expected annotation time for any dialog

\[
\text{expected annotation time} = t_{\text{draw}} + (1-p_i) * t_{\text{draw}}
\]
Agent 1: Model-based

Features (image, class, box)

Acceptance classifier

\[ p_{box} > \frac{t_{verify}}{t_{draw}} \]

Optimal strategy: sort boxes in order of acceptance probability; verify all boxes for which \( p > \frac{t_{verify}}{t_{draw}} \)
Proof

Algorithm IAD-Prob

1. \textbf{Input:} $B = \{b_1, \ldots, b_k\}$; $p(b_1), \ldots, p(b_k)$; $\epsilon > 0$
2. $\hat{S}_0 = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_k) = \text{sort}(A_0)$ by $p(b_i)$
3. $\hat{s}_1 \leftarrow 1$
4. $A_2 \leftarrow \emptyset$
5. while $p(\hat{s}_k) > \epsilon$, do
6. \hspace{1em} $A_2 \leftarrow A_2 \cup \{\hat{s}_k\}$
7. \hspace{1em} select action $V : \pi \rightarrow V$
8. \hspace{1em} select action $D : \pi \rightarrow D$
9. \hspace{1em} return sequence of actions $\pi$, sequence of boxes $A_2$

Theorem 1. If probabilities of acceptance $p(b_i)$ are known, the strategy of evoking a sequence of actions $V^mD$ defined by IAD-Prob in a sequence of boxes $A_n$ minimizes the annotation time, i.e. for all $m \in \{0, \ldots, n\}$ and for all box sequences $S_m$,

$$E[\tau(V^mD, A_n)] \leq E[\tau(V^mD, S_m)]$$

(1)

Sketch of the proof. The proof consists of two parts. First, we show that for any strategy $V^mD$, the best box sequence is obtained by sorting the available boxes by their probability of acceptance and using the first $m$ of them. Second, we show that the number of verification steps found by IAD-Prob, $k$, is indeed the optimal one.

We start by rewriting the expected episode length in closed form. For a strategy $V^mD$ and any sequence of boxes $S_m = \{s_1, s_2, \ldots, s_n\}$, we obtain

$$\tau(V^mD, S_m) = 1 + q(s_1)q(s_2) + \cdots - q(s_2)q(s_3) - \cdots - q(s_{n-1})q(s_n) - \epsilon = \sum_{i=1}^{n} \prod_{j=1}^{i} q(s_j)$$

(2)

Our first observation is that (2) is monotonically decreasing as a function of $q(s_1), \ldots, q(s_n)$. Consequently, the smallest value is obtained by selecting the set of $m$ boxes that have the smallest rejection probabilities. To prove that their optimal order is sorted in decreasing order, assume that $S_m$ is not sorted, i.e. there exist indices $i, j \in \{1, \ldots, n\}$ for which $q(s_i) > q(s_j)$. We compare the expected episode length of $S_m$ to that of a sequence $S_m$ in which $s_i$ and $s_{i+1}$ are at switched positions. Using (2) and noticing that many of the terms cancel out, we obtain

$$E[\tau(V^mD, S_m)] = E[\tau(V^mD, S_n)] - \epsilon \prod_{i=1}^{n-2} q(s_i) q(s_{i+1}) > 0$$

(3)

This shows that $S_m$ has strictly smaller expected episode length than $S_m$, so $S_m$ cannot have been the optimal order.

Consequently, for any strategy $V^mD$, the optimal sequence is to sort the boxes by decreasing probability of rejection, i.e. increasing acceptance probability. We denote it by $S^* = \{s_1, \ldots, s_n\}$.

Next, we show that the number $k$ of verification actions found by the IAD-Prob algorithm is optimal, i.e. $V^mD$ is better or equal to $V^mD$ (for any $m \geq k$). As we already know that the optimal box sequence for any strategy $V^mD$ is $S_m$, it is enough to show that

$$E[\tau(V^mD, S_m)] \leq E[\tau(V^mD, S_m)]$$

for all $m \in \{1, \ldots, k\}$, and

$$E[\tau(V^mD, S_m)] < E[\tau(V^mD, S_m)]$$

(4)

for all $m \in \{k, \ldots, n\}$. To prove these inequalities, we again make use of expression (2). For any $m \in \{1, \ldots, n-1\}$ we obtain

$$\tau(V^mD, S_m) = \tau(V^mD, S_n) - \epsilon \prod_{i=1}^{n-2} q(s_i) q(s_{i+1})$$

(5)

For $m \in \{1, \ldots, k\}$, we know that $p(s_m) > \epsilon / \prod_{i=1}^{m} q(s_i)$ by construction of the strategy. This is equivalent to $1 / \sum_{i=1}^{m} q(s_i) > \epsilon$. Consequently, (6) is non-negative in this case, and inequality (4) is confirmed. For $m \in \{k, \ldots, n\}$, we know $p(s_m) \leq \epsilon / \prod_{i=1}^{m} q(s_i)$ again by construction. Consequently, $\tau(V^mD, S_m) > \epsilon > 0$, which shows that (5) is non-positive in this case, confirming (5).

\hfill \Box
Agent 2: Reinforcement Learning

**Environment**

**State:** Features (image, class, box)

**Action:** ?? or 😐

**Reward:** \(-t_{\text{verify}}\) or \(-t_{\text{draw}}\)

**Agent:** Deep Q-learning
Experimental Settings

- Annotate PASCAL VOC 2007 trainval dataset
- Image-level labels are available
- 10% reserved for training the agent
- Faster-RCNN object detector [Ren NIPS 2015]
- Detector either weak (MIL on VOC07) or strong (fully supervised on VOC12)
- DQN (simplified) as a reinforcement learning algorithm [Mnih Nature 2015]
- RL is unstable for non-linear function approximation, so: experience replay, periodically updated Q-function
Different scenarios

- $\alpha = 0.5$
  - Desired quality: low
  - Detector strength: high
  - Drawing time: 26s
  - Train from boxes
  - [Su AAAI 12] ImageNet
  - V(verify)

- $\alpha = 0.7$
  - Desired quality: high
  - Detector strength: low
  - Drawing time: 7s
  - MIL from image labels
  - X-Click
  - D(raw)

- MIL from image labels
  - [Papadopoulos ICCV 17] X-Click

- Train from boxes
  - [Su AAAI 12] ImageNet
  - V(verify)
Different scenarios

Agent vs standard strategies
- **D**: Draw
- **V*D**: Box verification series

<table>
<thead>
<tr>
<th>Drawing technique</th>
<th>Slow drawing</th>
<th>Fast drawing</th>
<th>Fast drawing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detector</td>
<td>Weak detector</td>
<td>Weak detector</td>
<td>Strong detector</td>
</tr>
<tr>
<td>Quality level</td>
<td>$\alpha = 0.7$</td>
<td>$\alpha = 0.7$</td>
<td>$\alpha = 0.7$</td>
</tr>
<tr>
<td></td>
<td>$\alpha = 0.5$</td>
<td>$\alpha = 0.5$</td>
<td>$\alpha = 0.5$</td>
</tr>
<tr>
<td>D</td>
<td>25.50 ± 0.00</td>
<td>7.00 ± 0.00</td>
<td>7.00 ± 0.00</td>
</tr>
<tr>
<td>V*D</td>
<td>42.29 ± 0.07</td>
<td>31.82 ± 0.11</td>
<td>8.83 ± 0.09</td>
</tr>
<tr>
<td>Model-based agent</td>
<td>23.07 ± 0.23</td>
<td>6.81 ± 0.02</td>
<td>3.42 ± 0.18</td>
</tr>
<tr>
<td>RL agent</td>
<td>23.62 ± 0.38</td>
<td>6.83 ± 0.03</td>
<td>3.60 ± 0.07</td>
</tr>
</tbody>
</table>

Agent outperforms standard strategies in all scenarios
Different scenarios

Fixed strategies
- VD
- VVD
- VVVD

<table>
<thead>
<tr>
<th>Drawing technique</th>
<th>Slow drawing</th>
<th>Fast drawing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Weak detector</td>
<td>Weak detector</td>
</tr>
<tr>
<td></td>
<td>$\alpha = 0.7$</td>
<td>$\alpha = 0.5$</td>
</tr>
<tr>
<td>D</td>
<td>25.50 ± 0.00</td>
<td>25.50 ± 0.00</td>
</tr>
<tr>
<td>VD</td>
<td>23.01 ± 0.07</td>
<td>17.30 ± 0.07</td>
</tr>
<tr>
<td>VVD</td>
<td>23.79 ± 0.06</td>
<td>16.67 ± 0.06</td>
</tr>
<tr>
<td>VVVVD</td>
<td>24.67 ± 0.07</td>
<td>16.38 ± 0.07</td>
</tr>
<tr>
<td>V* D</td>
<td>42.29 ± 0.07</td>
<td>17.37 ± 0.07</td>
</tr>
</tbody>
</table>

No single fixed strategy works best in all scenarios
Different scenarios

Agent vs fixed strategies

<table>
<thead>
<tr>
<th>Drawing technique Detector Quality level</th>
<th>Slow drawing Weak detector $\alpha = 0.7$</th>
<th>Slow drawing Weak detector $\alpha = 0.5$</th>
<th>Fast drawing Weak detector $\alpha = 0.7$</th>
<th>Fast drawing Weak detector $\alpha = 0.5$</th>
<th>Fast drawing Strong detector $\alpha = 0.7$</th>
<th>Fast drawing Strong detector $\alpha = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>25.50 ± 0.00</td>
<td>25.50 ± 0.00</td>
<td>7.00 ± 0.00</td>
<td>7.00 ± 0.00</td>
<td>7.00 ± 0.00</td>
<td>7.00 ± 0.00</td>
</tr>
<tr>
<td>VD</td>
<td>23.01 ± 0.07</td>
<td>17.30 ± 0.07</td>
<td>7.62 ± 0.02</td>
<td>6.05 ± 0.02</td>
<td>3.45 ± 0.01</td>
<td>2.50 ± 0.01</td>
</tr>
<tr>
<td>VVD</td>
<td>23.79 ± 0.06</td>
<td>16.67 ± 0.06</td>
<td>8.92 ± 0.02</td>
<td>6.67 ± 0.02</td>
<td>3.48 ± 0.01</td>
<td>2.45 ± 0.01</td>
</tr>
<tr>
<td>VVVD</td>
<td>24.67 ± 0.07</td>
<td>16.38 ± 0.07</td>
<td>10.21 ± 0.02</td>
<td>7.32 ± 0.03</td>
<td>3.65 ± 0.02</td>
<td>2.48 ± 0.01</td>
</tr>
<tr>
<td>V*D</td>
<td>42.29 ± 0.07</td>
<td>17.37 ± 0.07</td>
<td>31.82 ± 0.11</td>
<td>11.46 ± 0.04</td>
<td>8.83 ± 0.09</td>
<td>3.18 ± 0.02</td>
</tr>
<tr>
<td>Model-based agent</td>
<td>23.07 ± 0.23</td>
<td>12.64 ± 1.29</td>
<td>6.81 ± 0.02</td>
<td>5.86 ± 0.04</td>
<td>3.42 ± 0.18</td>
<td>2.73 ± 0.08</td>
</tr>
<tr>
<td>RL agent</td>
<td>23.62 ± 0.38</td>
<td>16.30 ± 0.09</td>
<td>6.83 ± 0.03</td>
<td>5.89 ± 0.05</td>
<td>3.60 ± 0.07</td>
<td>2.66 ± 0.06</td>
</tr>
</tbody>
</table>

Agent is (almost) always best, *adapting* to scenario at hand
IAD with iteratively improving detector

In real annotation task we want to take advantage of the growing amount of data
Performance at @ IoU 0.7

- Fast drawing [Papadopoulos ICCV 17]
- Box verification series [Papadopoulos CVPR 16]
- Slow drawing [Su AAAI 12]

% of boxes annotated vs. annotation time (sec.)

Delivers detector at 98% mAP of always drawing
This talk

- Revisiting knowledge transfer for training object class detectors
  Uijlings, Popov, Ferrari
  CVPR 2018

- Learning intelligent dialogs for bounding box annotation
  Konyushkova, Uijlings, Lampert, Ferrari
  CVPR 2018

- Fluid Annotation: human-machine collaboration for full image annotation
  Andriluka, Uijlings, Ferrari,
  arXiv June 2018
Task: Full Image Annotation

• Annotate outline and class of every object and background region
• Extremely time consuming (19 min per image for COCO)

Lin et al., Microsoft COCO: common objects in context, ECCV 2014
Caesar et al., COCO-Stuff: Things and Stuff classes in context, CVPR 2018
Traditional Annotation System (e.g. LabelMe)

Input image
Manually draw a polygon ...
Hundreds of mouse-clicks later...

Traditional Annotation System (e.g. LabelMe)
Traditional Annotation System (e.g. LabelMe)

Hundreds of mouse-clicks later… Done!
Our Fluid Annotation System

Input image
Our Fluid Annotation System

Automatic initialization
Our Fluid Annotation System

tens of mouse clicks later… Done!
Design Principles

1. Strong Machine Learning Aid
2. Unified interface for full image annotation
3. Empower the annotator

Annotator focuses on what the machine does not already know
Method

apply Mask R-CNN

[He ICCV 2017]
Method

Segments (~1000) with Labels and detection score

- wall-concrete: 0.2
- tv: 0.7
- laptop: 0.9
- keyboard: 0.8
- knife: 0.2
- frisbee: 0.1
- blanket: 0.2
- clothing: 0.3
- door-stuff: 0.3
- bed: 0.4
- curtain: 0.1
- cat: 0.9
- metal: 0.3
Automatic Initialization

Most likely interpretation of scene

Present to annotator in simple interface
“Reorder” Action
“Reorder” Action
“Change label” Action
“Change label” Action
“Change label” Action
“Change label” Action
“Remove” Action
“Remove” Action
“Add” Action
“Add” Action
“Add” Action

no action, waiting ...
“Add” Action
Example Annotation Result
Results: Human Annotators

CLICK CLICK CLICK CLICK CLICK CLICK ... CLICK CLICK CLICK CLICK CLICK …

Recall

#micro-actions-per-image

LabelMe interface
Results: Human Annotators

Graph showing recall vs. number of micro-actions per image for two methods:
- LabelMe interface
- Fluid Annotation (human annotator)
Results: Human Annotators

![Graph showing recall over #micro-actions-per-image for different methods: LabelMe interface, Fluid Annotation (human annotator), and Fluid Annotation (simulation).]
Annotation time and Label Agreement

Lin et al., Microsoft COCO: common objects in context, ECCV 2014
Caesar et al., COCO-Stuff: Things and Stuff classes in context, CVPR 2018
Example Results
Example Results

COCO+Stuff original

Fluid Annotation

Polygons
Open Images V4 and Challenge

- 600 object classes
- **15.4M** bounding-boxes on 1.9M images
- 10x over existing datasets
- Complex images (average 8 boxes)
- Visual Relationship Detection annotations
- Challenge at ECCV 2018