High Level Computer Vision: Data Challenges

10th June, 2019

Bernt Schiele
Mario Fritz
Tribhuvanesh Orekondy
Motivation

• Models for Visual Recognition

• Supervised learning: requires data

• This lecture:
  ▸ What are some challenges?
  ▸ Methods to tackle them

Motivation

- Models for Visual Recognition
- Supervised learning: requires data
- Real-world
  
  Constant influx of real-world data
- Supervised learning world
  
  Capture + annotate real-world data
- Some challenges
  
  - Supervision/Annotation (Train-time)
  
  - Representations of data (Test-time)
Motivation

- Models for Visual Recognition
- Supervised learning: requires *data*
- Progress in datasets to learn visual recognition models?

Datasets: The Beginning

1972

Marr, 1976

Slide credit: "Recognizing and Learning Object Categories", Li Fei-Fei, Rob Fergus, Antonio Torralba
"CSE 590V: Computer vision seminar", Neeraj Kumar and Bryan Russell. Source.
The 2000s

Caltech 101
Fei-Fei, Fergus, Perona, 2004

Caltech 256
Griffin, Holub, Perona, 2007

Slide credit: "Recognizing and Learning Object Categories", Li Fei-Fei, Rob Fergus, Antonio Torralba
"CSE 590V: Computer vision seminar", Neeraj Kumar and Bryan Russell. Source.
The 2000s

Slide credit: "Recognizing and Learning Object Categories", Li Fei-Fei, Rob Fergus, Antonio Torralba
"CSE 590V: Computer vision seminar", Neeraj Kumar and Bryan Russell. Source.
Complex Tasks

Semantic Segmentation

Classification + Localization

Object Detection

Instance Segmentation

GRASS, CAT, TREE, SKY

No objects, just pixels

CAT

Single Object

DOG, DOG, CAT

Multiple Object

DOG, DOG, CAT
Rise of the Modern Datasets

**ImageNet**
14M images, 20K categories

**ILSVRC**

**MS-COCO**
Segmentation. 1.2M images, 1K categories

**Cityscapes**
Segmentation. 123K images, 80 categories

Segmentation. 5K + 20K images, 30 categories

**Source**
Canziani et al.

Challenges

- Train-time
  - Learning with Less Supervision
  - Active Learning, Efficient Clicking, ...

- Test-time
  - Domain Shift
  - Out-of-Distribution, Explainability, ...
Challenges

• Train-time
  ▶ Learning with Less Supervision
  ▶ Active Learning, Efficient Clicking, ...

• Test-time
  ▶ Domain Shift
  ▶ Out-of-Distribution, Explainability, ...
Learning with Less Supervision

- Supervised Learning. Train/test data: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$


Crowd-sourcing Annotation

Slide credit: "Recognizing and Learning Object Categories", Li Fei-Fei, Rob Fergus, Antonio Torralba
"CSE 590V: Computer vision seminar", Neeraj Kumar and Bryan Russell. Source.
Learning with Limited Data

- Supervised Learning. Train/test data: \( \mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N} \)

- Dataset collection cost
  - Images = $
  - Annotations = $$$ \rightarrow \text{bottleneck}

- You get what you pay for:

- Can we do more with less supervision?

\( F \) \( (w = \text{rand}) \) \( \text{supervised} \) \( N \gg 0 \) \( \{(x_i, y_i)\} \) \( \overset{\text{Train data}}{\rightarrow} \) \( \overset{\text{Train for Task A}}{\rightarrow} \overset{\text{Test on Task A}}{\rightarrow} \)

\( w^* \)

\( y = F_{w^*}(x) \)
$F$ \hspace{1cm} \textit{supervised} \hspace{1cm} N \gg 0 \rightarrow \{(x_i, y_i)\}_{i=1}^{N} \rightarrow \text{Train for Task A} \rightarrow \text{Test on Task A}

\begin{align*}
\text{Train data} & \quad \text{Test on Task A} \\
(x_i, y_i) & \quad y = F_{w^*}(x)
\end{align*}

- Traditionally: \{supervised, semi-supervised, unsupervised\}
- But: Many other types of supervision
ImageNet + Deep Learning

- Image Retrieval
- Detection (RCNN)
- Segmentation (FCN)
- Depth Estimation
- ...

Following slides adapted from "Unsupervised Visual Representation Learning by Context Prediction" by Carl Doersch
ImageNet + Deep Learning

Materials?
Geometry?
Parts?
Boundaries?
Pose?

Beagle
Train data $\{(x_i, y_i)\}_{i=1}^{N}$

$F$ (w = rand)

supervised

Train on $\{(x_i, z_i)\}$ → Task B

pretrain

Train for $\tilde{w}$

self-supervised

unsupervised

weakly-supervised

synthetic training data

semi-supervised

zero/few-shot

Zero/few labels for some classes

Train for Task A

$w^*$

Test on Task A

$y = F_{w^*}(x)$

ImageNet Challenge

Task B

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

Test A


Train data
\((x_i, y_i)\) for Task A

Train for Task A

Test on Task A

\(y = F_{w^*}(x)\)

\(F\)

\((w = \text{rand})\)

\(\tilde{w}\)

\(\{x_i, z_i\}\) → Task B

\(\{x_i, y_i(x_i)\}\) → Task C

\(\{x_i, x_i\}\) → Task D

supervised

\(N \gg 0\)

\(\{x_i, y_i\}\) → Task A

pretrain

self-supervised*

unsupervised

weakly-supervised

synthetic training data

semi-supervised

zero/few-shot

Zero/few labels for some classes

• Task B: requires labels \(z_i\)

• Task C: Can we learn from context of an image \(y(x_i)\)?

• Exploits inherent structure


Self-supervised Learning
Context as Supervision
[Collobert & Weston 2008; Mikolov et al. 2013]

house, where the professor lived without his wife and child; or so he said jokingly sometimes: “Here’s where I live. My house.” His daughter often added, without resentment, for the visitor’s information, “It started out to be for me, but it’s really his.” And she might reach in to bring forth an inch-high table lamp with fluted shade, or a blue dish the size of her little fingernail, marked “Kitty” and half full of eternal wine, but she was sure to replace these, after they had been admired, pretty near exactly where they had been. The little house was very orderly, and just big enough for all it contained, though to some tastes the bric-à-brac in the parlor might seem excessive. The daughter’s preference was for the store-bought gimmicks and appliances, the toasters and carpet sweepers of Lilliput, but she knew that most adult visitors would
In this paper we present several extensions of the original Skip-gram model. We show that substituting word vectors with phrase vectors results in more training examples and thus can lead to a higher accuracy, at the expense of longer training time.

Formally, given a sequence of words $w_t, w_{t+1}, \ldots, w_T$, the training objective of the Skip-gram model is to maximize the average log probability of predicting the nearby words:

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t)$$

where $c$ is the size of the training context (which can be a function of $T$).

The Skip-gram model architecture. The training objective is to find word vectors $\mathbf{w}_t$ such that is good at predicting the surrounding words in a sentence or a document. The extension from word based to phrase based models is relatively simple. First we identify a large number of frequent words, compared to more complex representations that are useful for representing idiomatic phrases that are not composed naturally.

We present a simplification that eliminates the need for storing the whole phrases as individual tokens. The extension from word based to phrase based models is relatively simple. First we identify a large number of frequent words, compared to more complex representations that are useful for representing idiomatic phrases that are not composed naturally.

Finally, we describe another interesting property of the Skip-gram model. We show that substituting word vectors with phrase vectors improves accuracy of the representations of less frequent words. In addition, we present a simplified variant of Noise Contrastive Estimation (NCE) [4] for training the Skip-gram model that results in faster training and better vector representations for frequent words, compared to more complex representations.

Word2Vec	Mikolov’13

![Diagram of the Skip-gram model](Image 261x-1456 to 286x-1454)
Context Prediction for Images
Semantics from a non-semantic task
Relative Position Task

8 possible locations

Randomly Sample Patch
Sample Second Patch

CNN

Classifier
Patch Embedding

Note: connects *across* instances!
Siamese Architecture

- Chopra’05 “Learning a Similarity Metric Discriminatively, with Application to Face Verification”
- Bromley’94 “Signature verification using a siamese time delay neural”
Avoiding Trivial Shortcuts

Include a gap

Jitter the patch locations
A Not-So “Trivial” Shortcut
What is learned?

Input  | Ours  | Random Initialization | ImageNet AlexNet
---     | ---   | ---                  | ---               

Images of various objects are shown, comparing the learned representations with random initialization and ImageNet AlexNet.
Still don’t capture everything

You don’t always need to learn!
Visual Data Mining

Via Geometric Verification

Simplified from [Chum et al 2007]
Mined from Pascal VOC2011
Train on $\{ (x_i, z_i) \}$ \rightarrow Task B

**pretrain**

Train on $\{ (x_i, y_i(x_i)) \}$ \rightarrow Task C

**self-supervised**

Train data

$\{(x_i, y_i)\}_{i=1}^{N}$

Train for Task A

$w^*$

Test on Task A

$y = F_{w^*}(x)$

• Task B: requires labels $z_i$

• Task C: Can we learn from context of an image $y(x_i)$?

• Exploits inherent structure


Pre-Training for R-CNN

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

Pre-train on relative-position task, w/o labels

[Girshick et al. 2014]
VOC 2007 Performance
(pretraining for R-CNN)

% Average Precision

ImageNet Labels

- No Rescaling
- Krähenbühl et al. 2015
- VGG + Krähenbühl et al.

68.6
56.8
54.2
46.3
51.1
40.7
42.4
45.6

[Krähenbühl, Doersch, Donahue & Darrell, “Data-dependent Initializations of CNNs”, 2015]
Capturing Geometry?
Train data
\(\{(x_i, y_i)\}_{i=1}^{N}\)

\(w = \text{rand}\)

\(F\)

\(\superscript{\text{supervised}}{N} \gg 0\)

Train for
\(\text{Task A}\)

Test on
\(\text{Task A}\)

\(w^*\)

\(y = F_{w^*}(x)\)

\(\superscript{\text{pretrain}}{(x_i, z_i)}\)

\(\superscript{\text{self-supervised}}{(x_i, y_i(x_i))}\)

\(\superscript{\text{unsupervised}}{(x_i, x_i)}\)

\(\superscript{\text{weakly-supervised}}{(x_i, y_i^{\text{weak}})}\)

\(\superscript{\text{synthetic training data}}{(\bar{x}_i, y_i)}\)

\(\superscript{\text{semi-supervised}}{(x_i, y_i)} \cup \{x_i\}\)

\(\superscript{\text{zero/few-shot}}{\text{Zero/few labels for some classes}}\)

Task C = context prediction
bottom-right

\(x_i\)

\(y(x_i)\)

Task C = inpainting

Train data

\( \{(x_i, y_i)\}_{i=1}^{N} \)

\( F \)

\( w = \text{rand} \)

\( \text{supervised} \)

\( N \gg 0 \)

\( \text{Train for Task A} \)

\( w^* \)

\( \text{Test on Task A} \)

\[ y = F_{w^*}(x) \]

**Task D**

- L2 Loss function:
  \[ \|x - \hat{x}\|^2 \]
- Encoder: 4-layer conv
- Decoder: 4-layer upconv
- Input data
- Features \( z \)
- Decoder
- Reconstructed data

**Autoencoders**

- \( F \)
- \( w \)
- \( \tilde{w} \)
- Weakly-supervised
- Synthetic training data
- Semi-supervised
- Zero/few-shot

\( \{(x_i, y_i^{\text{weak}})\} \)

\( \{(\tilde{x}_i, y_i)\} \)

\( \{(x_i, y_i)\} \cup \{x_i\} \)

Zero/few labels for some classes


Fei-Fei Li, Justin Johnson, Serena Yeung. Stanford cs231n
Train data \( \{(x_i, y_i)\}_{i=1}^N \)

Train for Task A

Test on Task A

\( w^* \)

\( y = F_{w^*}(x) \)

---

**Task A**

- **Train on**
  - **supervised**
    \( \{ (x_i, z_i) \} \rightarrow \text{Task B} \)
  - **pretrain**
    \( \{ (x_i, y_i(x_i)) \} \rightarrow \text{Task C} \)
  - **self-supervised**
    \( \{ (x_i, x_i) \} \rightarrow \text{Task D} \)

- **weakly-supervised**
  \( \{ (x_i, y_i^{\text{weak}}) \} \rightarrow \text{Task A} \)

---

- **synthetic training data**
- **semi-supervised**
- **zero/few-shot**

---

Train data

\( \{ (x_i, y_i) \}_{i=1}^{N} \)

**Train for Task A**

\( w^{*} \)

**Test on Task A**

\( y = F_{w^{*}}(x) \)

- **Train on**
  - \( \{ (x_i, z_i) \} \) → Task B
  - \( \{ (x_i, y_i(x_i)) \} \) → Task C
  - \( \{ (x_i, x_i) \} \) → Task D

**supervised**

\( F^{(w = \text{rand})} \)

**self-supervised**

**unsupervised**

**weakly-supervised**

- **synthetic training data**
- **semi-supervised**
- **zero/few-shot**
  - Zero/few labels for some classes

**Task A**

- Image
- Ground truth
- Semi supervised \( x \cap G \)
- Fully supervised


Weakly-supervised Learning: Guided Labeling

Task & Motivation

Learning to segment objects from image label annotations.

- Cheaper than full supervision.
- Humans can do.

Training: Image Labels

→ Final task: Semantic Labelling
1. Get discriminative object locations from a image-level classifier [1,2] (seed).
2. Image labels alone do not give full object extent information (e.g. train and rail); we propose to exploit class-agnostic image-level saliency (saliency).
3. Combine the two sources of information (guide labels).
4. Refine the labelling by training a segmenter (e.g. DeepLab [4]) with the guide labels.
1. Seed : Encode Image Labels

Discriminative object locations from image-level classifiers.

- How to get localized information from image level annotation (weakly supervised)?
  - Image gradient
  - Global Average Pooling (GAP)
Seed: Image Gradient
Seed: Global Average Pooling (GAP)


- Weakly supervised

  \[ x_i \quad y_i^{\text{weak}} \quad y_i \]

  "brushing teeth"

  "cutting trees"
Global Average Pooling and Class Activation Maps

\[ F_k = \sum_{x,y} f_k(x, y) \]

\[ S_c = \sum_k w_k^c \sum_{x,y} f_k(x, y) = \sum_{x,y} \sum_k w_k^c f_k(x, y) \]

\[ M_c(x, y) = \sum_k w_k^c f_k(x, y) \]
Comparison to Gradient and Qualitative Results

- Figures show class activation maps (CAM) from CNNs and the class-specific saliency map from the backpropagation methods.
- Images represent different scene categories from various datasets.
- The CAMs highlight important regions in the images for discriminative localization.
- Tables summarize accuracy results for different approaches and datasets.
1. Seed: Encode Image Labels

Class specific discriminative locations from an image level classifier.

- **Data**: Pascal images + image labels.
- **Model**: fully convolutional network + global average pooling (GAP) [1,2].

(Left) Class-averaged precision-recall curves for GAP and Gradient based seeds. (Right) Qualitative results of GAP variants.
1. Get discriminative object locations from a image-level classifier \([1,2]\) (seed).
2. Image labels alone do not give full object extent information (e.g. train and rail); we propose to exploit class-agnostic image-level saliency (saliency).
3. Combine the two sources of information (guide labels).
4. Refine the labelling by training a segmenter (e.g. DeepLab [4]) with the guide labels.
2. Saliency : Encode “Objectness” Prior

Image labels alone are insufficient statistic for semantic segmentation. (saliency)

• Data: 11k MSRA single-object images with boxes [3]. Only non-Pascal classes are used for class-genericity of the mask.

• Model: DeepLab [4].

Foreground mask of generic object class.

Image labels alone do not give full object extent information (e.g. train and rail); we propose to exploit class-agnostic image-level saliency. (saliency).
3. Guide Labels : Seed + Saliency

Combination algorithm:

i. Break seed and saliency into connected components.

ii. If seeds touch saliency: diffuse seeds inside saliency with dense CRF.

iii. If seed is alone, label as FG; If saliency is alone, label as BG.
4. Semantic Segmentation Result & Comparison

<table>
<thead>
<tr>
<th>Method</th>
<th>Data</th>
<th>Val mIoU</th>
<th>Test mIoU</th>
<th>FS%</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIL-FCN</td>
<td>I+P</td>
<td>25.0</td>
<td>25.6</td>
<td>36.5</td>
</tr>
<tr>
<td>DCSM</td>
<td>I+P</td>
<td>44.1</td>
<td>45.1</td>
<td>64.2</td>
</tr>
<tr>
<td>SEC</td>
<td>I+P</td>
<td>50.7</td>
<td>51.7</td>
<td>73.5</td>
</tr>
<tr>
<td>STC</td>
<td>I+P+S+E40k</td>
<td>49.8</td>
<td>51.2</td>
<td>72.8</td>
</tr>
<tr>
<td>CheckMask</td>
<td>I+P+µ</td>
<td>51.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MicroAnno</td>
<td>I+P+µ</td>
<td>51.9</td>
<td>53.2</td>
<td>75.7</td>
</tr>
<tr>
<td>GuideLabel</td>
<td>I+P+S</td>
<td>55.7</td>
<td>56.7</td>
<td>80.6</td>
</tr>
<tr>
<td>DeepLabv1 [4]</td>
<td>I+Pfull</td>
<td>67.6</td>
<td>70.3</td>
<td>100</td>
</tr>
</tbody>
</table>

- Reach 80% of the fully supervised performance.
- Better saliency model will further improve the result; oracle saliency gives 61.8 mIoU.
Train data

Train for Task A

Test on Task A

$w^*$

$y = F_{w^*}(x)$

Task A

Train on

Train for pretrain

$(x_i, z_i)$ → Task B

self-supervised

$(x_i, y_i(x_i))$ → Task C

unsupervised

$(x_i, x_i)$ → Task D

weakly-supervised

$(x_i, y_i^{\text{weak}})$ →

synthetic training data

semi-supervised

$(x_i, y_i)$ →

zero/few-shot

Zero/few labels for some classes


Train data

Train for Task A

Test on Task A

\(\tilde{w}\)

\(\{(x_i, y_i)\}_{i=1}^{N}\)

Train on Task B

\(\{(x_i, z_i)\}\)

Train for Task C

\(\{(x_i, y_i(x_i))\}\)

Train for Task D

\(\{(x_i, x_i)\}\)

Supervised

\(w^*\)

Zero/few-shot

\(y = F_{w^*}(x)\)

- Train data

\[ \{ (x_i, y_i) \}_{i=1}^{N} \]

- \( F \) (\( w = \text{rand} \))

- \( \tilde{w} \)

- Train on

\[ \{ (x_i, z_i) \} \]

- Task B

- Train for

\[ \{ (x_i, y_i(x_i)) \} \]

- Task C

- \( \{ (x_i, x_i) \} \)

- Task D

- \( \text{supervised} \)

- \( \text{pretrain} \)

- \( \text{self-supervised} \)

- \( \text{unsupervised} \)

- \( \text{weakly-supervised} \)

- \( \text{synthetic training data} \)

- \( \text{semi-supervised} \)

- Zero/few-shot

\[ \{ (x_i, y_i) \} \cup \{ x_i \} \]

\[ y = F_{w^*}(x) \]

Train data

Train on
\(\{(x_i, z_i)\}\) → Task B

Train for
\(\{(x_i, y_i(x_i))\}\) → Task C

Train on (\(w = \text{rand}\))
\(\left\{ (x_i, x_i) \right\} \) → Task D

\(\text{supervised} \quad N \gg 0 \)

\(\text{pretrain} \quad \tilde{w}\)

\(\text{self-supervised}
\{(x_i, y_i(x_i))\}\) → Task C

\(\text{unsupervised}
\left\{ (x_i, x_i) \right\} \) → Task D

\(\text{weakly-supervised}
\left\{ (x_i, y_i^{\text{weak}}) \right\} \) →

\(\text{synthetic training data}
\left\{ (\tilde{x}_i, y_i) \right\} \) →

\(\text{semi-supervised}
\left\{ (x_i, y_i) \right\} \cup \left\{ x_i \right\} \) →

\(\text{zero/few-shot}
\text{Zero/few labels for some classes}

\(\text{Test on Task A}
\quad w^* \quad y = F_{w^*}(x)\)

Challenges

• Train-time
  ▶ Learning with Less Supervision
  ▶ Active Learning, Efficient Clicking, ...

• Test-time
  ▶ Domain Shift
  ▶ Out-of-Distribution, Explainability, ...
Many other challenges

- **Supervised Learning**

  - **raw unlabeled data**
    \[ x_1, x_2, x_3, \ldots \]

  - **supervised learner**
    induces a classifier

  - **expert / oracle**
    analyzes experiments to determine labels

Many other challenges

- Supervised Learning

Many other challenges

- Supervised Learning

Many other challenges

- Active Learning: Which data to label?
Many other challenges

- Active Learning: Which data to label?
Many other challenges

- Active Learning: Which data to label?
Many other challenges

- Active Learning: Which data to label?
Many other challenges

- Active Learning: Which data to label?
Many other challenges

• Active Learning: Which data to label?

• Uncertainty-based approaches: Select sample $x^*$ that the current model $\theta$ is most uncertain about

• Some typical measures:

  • **Least Confident:** $x_{LC}^* = \arg\max_x 1 - P_\theta(\hat{y}|x)$
    where $\hat{y}$ is the most probable label for $x$ under the current model $\theta$

  • **Smallest Margin:** $x_{SM}^* = \arg\min_x P_\theta(y_1|x) - P_\theta(y_2|x)$
    $y_1, y_2$ are the two most probable labels for $x$ under the current model

  • **Label Entropy:** choose example whose label entropy is maximum
    $$x_{LE}^* = \arg\max_x - \sum_{i} P_\theta(y_i|x) \log P_\theta(y_i|x)$$
    where $y_i$ ranges over all possible labels

Many other challenges

- Active Learning: Which data to label?
- Efficient clicking: How to minimize #clicks for annotation?
Many other challenges

- Active Learning: Which data to label?
- Efficient clicking: How to minimize #clicks for annotation?
- Label Noise: How to learn in presence of noisy labels?
- ...
Challenges

- **Train-time**
  - Learning with Less Supervision
  - Active Learning, Efficient Clicking, ...

- **Test-time**
  - Domain Shift
  - Out-of-Distribution, Explainability, ...
Datasets for Visual Recognition

- In Machine Learning
  Dataset is the world

- In Visual Recognition
  Dataset is a *representation* of the world

Want: dataset = good representation of the world

When training models (e.g., CNNs), we are training on/for this representation

Slides adapted from: "Unbiased Look at Dataset Bias", Torralba and Efros
Game 1: Name That Dataset

People: 75% correct

__ Caltech 101
__ Caltech 256
__ MSRC
__ UIUC cars
__ Tiny Images
__ Corel
__ PASCAL 2007
__ LabelMe
__ COIL-100
__ ImageNet
__ 15 Scenes
__ SUN’09
Game 1: Name That Dataset

People: 75% correct
SVMs: 39% correct

- If datasets similarly represent the visual world, this should not be possible

- Do models learn dataset-specific representations?

Slides adapted from: "Unbiased Look at Dataset Bias", Torralba and Efros
# Game 2: Measuring Dataset Bias

<table>
<thead>
<tr>
<th>task</th>
<th>Train on:</th>
<th>Test on:</th>
<th>SUN09</th>
<th>LabelMe</th>
<th>PASCAL</th>
<th>ImageNet</th>
<th>Caltech101</th>
<th>MSRC</th>
<th>Self</th>
<th>Mean others</th>
<th>Percent drop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUN09</td>
<td>28.2</td>
<td>29.5</td>
<td>16.3</td>
<td>14.6</td>
<td>16.9</td>
<td>21.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LabelMe</td>
<td>14.7</td>
<td>34.0</td>
<td>16.7</td>
<td>22.9</td>
<td>43.6</td>
<td>24.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PASCAL</td>
<td>10.1</td>
<td>25.5</td>
<td>35.2</td>
<td>43.9</td>
<td>44.2</td>
<td>39.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ImageNet</td>
<td>11.4</td>
<td>29.6</td>
<td>36.0</td>
<td>57.4</td>
<td>52.3</td>
<td>42.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Caltech101</td>
<td>7.5</td>
<td>31.1</td>
<td>19.5</td>
<td>33.1</td>
<td>96.9</td>
<td>42.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MSRC</td>
<td>9.3</td>
<td>27.0</td>
<td>24.9</td>
<td>32.6</td>
<td>40.3</td>
<td>68.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean others</td>
<td>10.6</td>
<td>28.5</td>
<td>22.7</td>
<td>29.4</td>
<td>39.4</td>
<td>34.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Severe**

- **Undesirable:** Want models to generalize across other "representations" of the visual world

Selection Bias: Sampling "Paris" images

Flickr

Bing

Google Street View

Aloysha Efros
Bias sources

- **Sampling Bias**: People like to take photos on vacation

- **Photographer Bias**: People want photographs to be interesting
Bias sources

- Social Bias: People like special moments

"100 Special Moments" by Jason Salavon

Slides adapted from: "Unbiased Look at Dataset Bias", Torralba and Efros
In the Deep Learning Era

Train on Cityscapes

Test on Cityscapes


Slides adapted from: Saenko, Kate. "Domain adaptation for deep learning". Source.
In the Deep Learning Era

(Trained on Cityscapes.)

Test on San Francisco dashcam.


Slides adapted from: Saenko, Kate. "Domain adaptation for deep learning". Source.
Domain Adaptation

- Broader goal: Adapt from **source** domain to **target** domain
- Want: Neural network that works well on **target** domain

Slides adapted from: Saenko, Kate. "Domain adaptation for deep learning". Source. 
Domain Adaptation

- Typical assumption

Source Domain $\sim P_S(X, Y)$
- lots of labeled data

$D_S = \{(x_i, y_i), \forall i \in \{1, \ldots, N\}\}$

Target Domain $\sim P_T(Z, H)$
- unlabeled or limited labels

$D_T = \{(z_j, ?), \forall j \in \{1, \ldots, M\}\}$

Slides adapted from: Saenko, Kate. "Domain adaptation for deep learning". Source.
When trained on source only, feature distributions do not match.

\[ S(f) = \{ G_f(x; \theta_f) | x \sim S(x) \} \]
\[ T(f) = \{ G_f(x; \theta_f) | x \sim T(x) \} \]
Domain Shift

When trained on source only, feature distributions do not match.

\[
S(f) = \{ G_f(x; \theta_f) | x \sim S(x) \} \\
T(f) = \{ G_f(x; \theta_f) | x \sim T(x) \}
\]

Domain Shift

When trained on source only, feature distributions do not match.

\[
S(f) = \{ G_f(x; \theta_f) | x \sim S(x) \}
\]

\[
T(f) = \{ G_f(x; \theta_f) | x \sim T(x) \}
\]
Domain Shift

When trained on source only, feature distributions do not match.

Our goal is to get this:

Domain Classifier

- Computes \( d = G_d(f; \theta_d) \)
- Is trained to predict 0 for source and 1 for target
- Therefore, the domain loss

is low for is higher for

Computes $d = G_d(f; \theta_d)$
- Is trained to predict 0 for source and 1 for target
- Therefore, the domain loss is low for is higher for

Domain Classifier

- Computes $d = G_d(f; \theta_d)$
- Is trained to predict $0$ for source and $1$ for target
- Therefore, the domain loss is low for and is higher for

How to train? Gradient Reversal Layer (GRL)

Let's try standard backpropagation. Emerging features are:
- Discriminative (i.e. good for predicting $y$)
- Domain-discriminative (i.e. good for predicting $d$)

How to train? Gradient Reversal Layer (GRL)

Let’s try standard backpropagation. Emerging features are:

- Discriminative (i.e. good for predicting $y$)
- Domain-discriminative (i.e. good for predicting $d$)

How to train? Gradient Reversal Layer (GRL)

Let’s now inject the **Gradient Reversal Layer**:

- Copies data without change at \( fprop \)
- Multiplies deltas by \(-\lambda\) at \( bprop \)
How to train? Gradient Reversal Layer (GRL)

Emerging features are now:

- Discriminative (i.e. good for predicting \( y \))
- Domain-invariant (i.e. not good for predicting \( d \))
Saddle Point Interpretation

Our objective is

\[
E(\theta_f, \theta_y, \theta_d) = \sum_{i=1..N} L_y^i(\theta_f, \theta_y) - \lambda \sum_{i=1..N} L_d^i(\theta_f, \theta_d)
\]

The backpropagation converges to a **saddle point**: 

\[
(\hat{\theta}_f, \hat{\theta}_y) = \arg \min_{\theta_f, \theta_y} E(\theta_f, \theta_y, \hat{\theta}_d)
\]

\[
\hat{\theta}_d = \arg \max_{\theta_d} E(\hat{\theta}_f, \hat{\theta}_y, \theta_d)
\]

Similar idea in **Generative Adversarial Networks** *(Goodfellow et al., 2014)*.
Workflow Overview

1. Train feature extractor + label predictor on source
2. Train feature extractor + domain classifier on source + target
3. Use feature extractor + label predictor at test time

Workflow Overview

1. Train **feature extractor** + **label predictor** on **source**
2. Train **feature extractor** + **domain classifier** on **source** + **target**
3. Use **feature extractor** + **label predictor** at test time

Workflow Overview

1. Train feature extractor + label predictor on source
2. Train feature extractor + domain classifier on source + target
3. Use feature extractor + label predictor at test time

Workflow Overview

1. Train **feature extractor + label predictor** on **source**
2. Train **feature extractor + domain classifier** on **source + target**
3. Use **feature extractor + label predictor** at **test time**
The Office Dataset

**Source:** office objects on white background

**Target:** photos of office objects taken by a webcamma

# The Office Dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Source Target</th>
<th>Amazon Webcam</th>
<th>DSLR Webcam</th>
<th>Webcam DSLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFK(PLS, PCA) (Gong et al., 2013)</td>
<td></td>
<td>.197</td>
<td>.497</td>
<td>.631</td>
</tr>
<tr>
<td>SA (Fernando et al., 2013)</td>
<td></td>
<td>.450</td>
<td>.648</td>
<td>.699</td>
</tr>
<tr>
<td>DLID (S. Chopra &amp; Gopalan, 2013)</td>
<td></td>
<td>.519</td>
<td>.782</td>
<td>.899</td>
</tr>
<tr>
<td>DDC (Tzeng et al., 2014)</td>
<td></td>
<td>.618</td>
<td>.950</td>
<td>.985</td>
</tr>
<tr>
<td>DAN (Long &amp; Wang, 2015)</td>
<td></td>
<td>.685</td>
<td>.960</td>
<td>.990</td>
</tr>
<tr>
<td>Source only</td>
<td></td>
<td>.642</td>
<td>.961</td>
<td>.978</td>
</tr>
<tr>
<td>Proposed Approach</td>
<td></td>
<td><strong>.730</strong></td>
<td><strong>.964</strong></td>
<td><strong>.992</strong></td>
</tr>
</tbody>
</table>

**Protocol:** all of the methods above use
- all available labeled source samples
- all available unlabeled target samples

Synthetic to Real

Source: rendered numbers

Target: SVHN

Accuracy

0.92

0.91

0.87

0.86

Lower bound

SA

Ours

Upper bound

Larger Domain Shift

Source: SVHN

Target: MNIST

Challenges

- **Train-time**
  - Learning with Less Supervision
  - Active Learning, Efficient Clicking, ...

- **Test-time**
  - Domain Shift
  - Out-of-Distribution, Explainability, ...
Many other challenges

- Confidence Calibration

Many other challenges

- Confidence Calibration
- High-confidence predictions far away from training data

Many other challenges

- Confidence Calibration

- High-confidence predictions far away from training data

- High-confidence predictions close to test image (Adversarial example)

\[
\begin{align*}
x & \quad \text{“panda”} \quad 57.7\% \text{ confidence} \\
+ 0.007 \times \text{sign}(\nabla_x J(\theta, x, y)) & \quad \text{“nematode”} \quad 8.2\% \text{ confidence} \\
= \quad x + \varepsilon \text{sign}(\nabla_x J(\theta, x, y)) & \quad \text{“gibbon”} \quad 99.3\% \text{ confidence}
\end{align*}
\]

Many other challenges

- Confidence Calibration
- High-confidence predictions far away from training data
- High-confidence predictions close to test image (Adversarial example)
- Out-of-distribution predictions