Webly Supervised Learning of Convolutional Networks

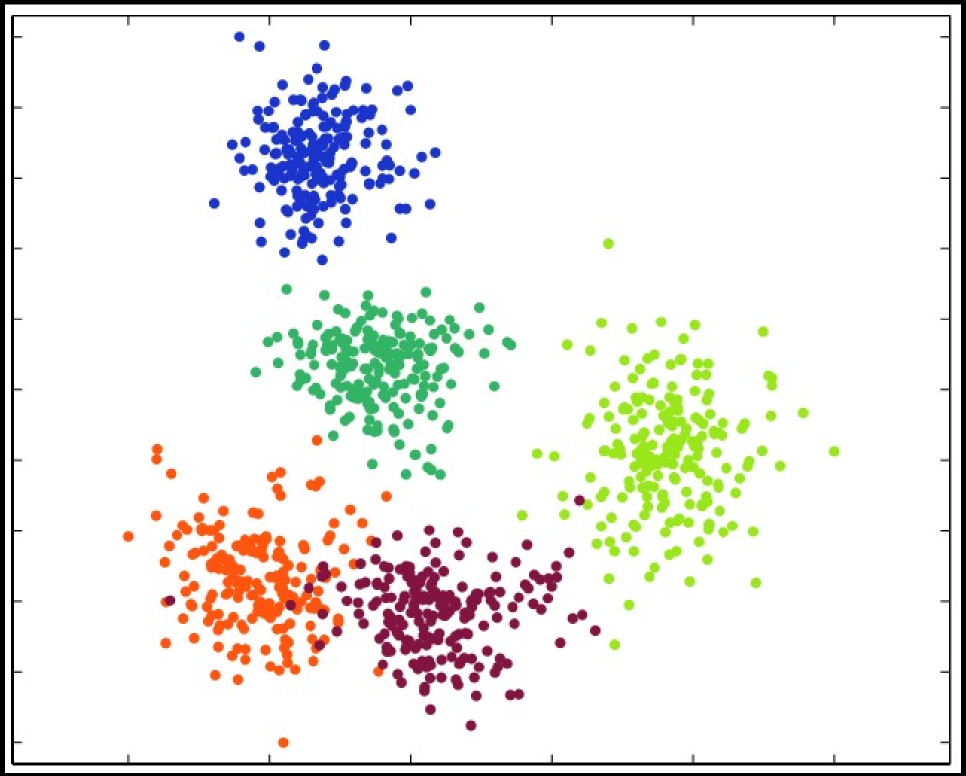
Xinlei Chen, Abhinav Gupta
ICCV, 2015

LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, Jianxiong Xiao
arXiv, 2015

Presenter: Igor Janjic
11/12/2015

Machine Learning



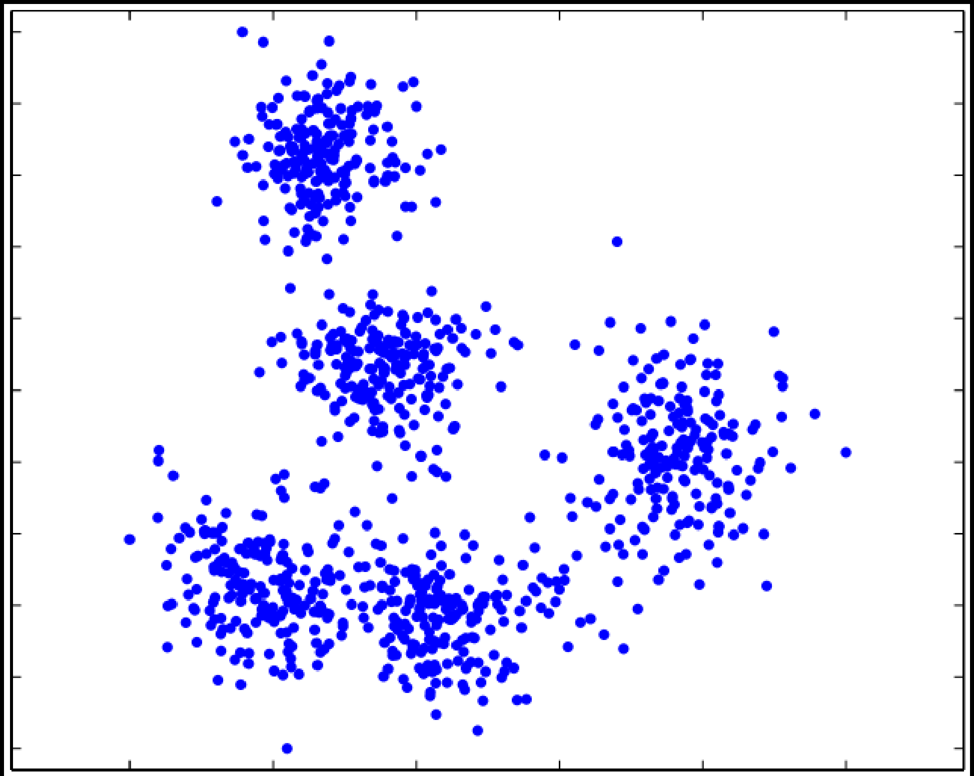


Image source: Prof. Kai Arras
(Social Robotics Lab)

Supervised ⇐⇒ **Semi-Supervised** ⇐⇒ **Unsupervised**

(Weakly Supervised)

Why is unsupervised learning important?

Answer: For learning general representations

1. Most data is unlabeled or weakly labeled

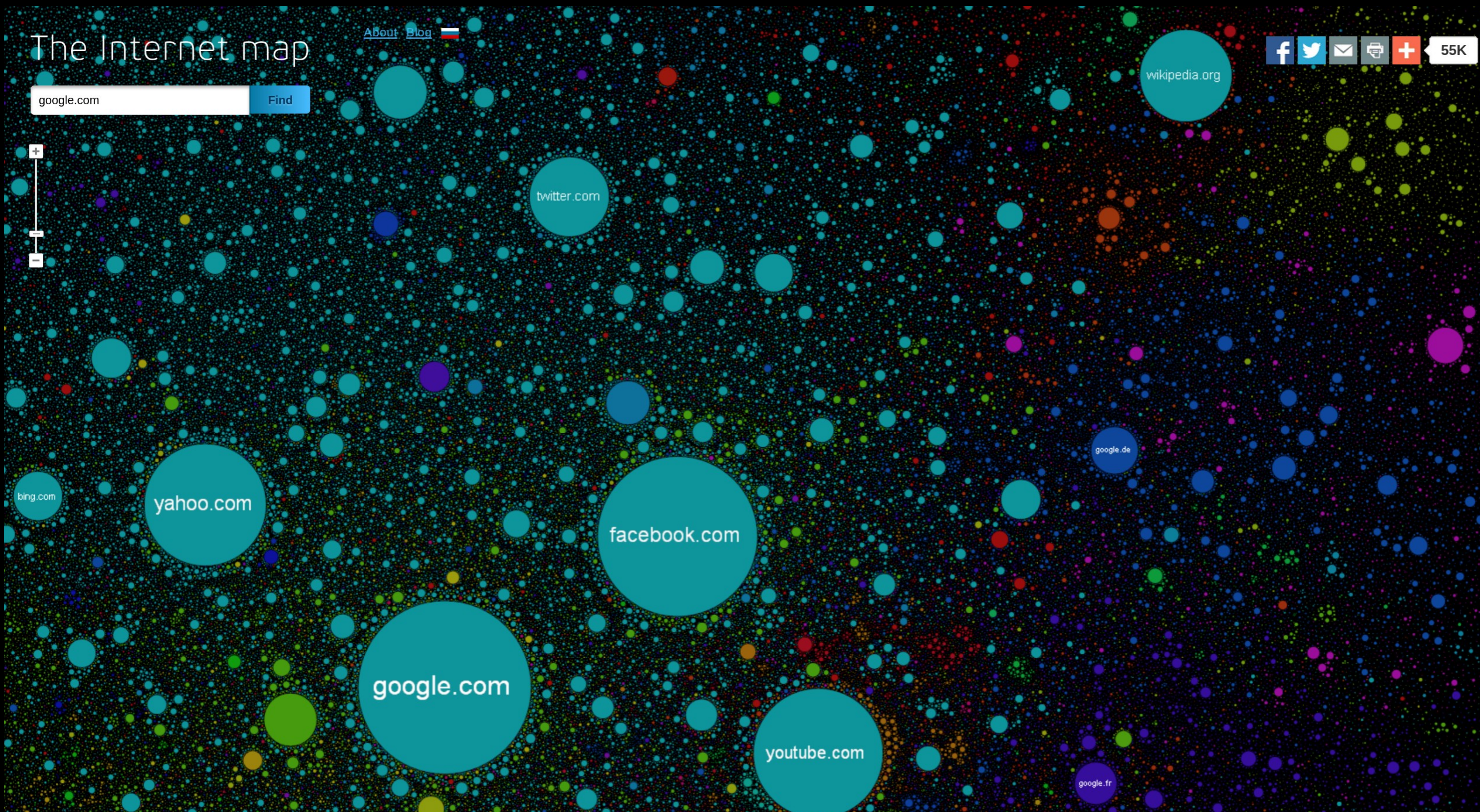


Image source: internet-map.net

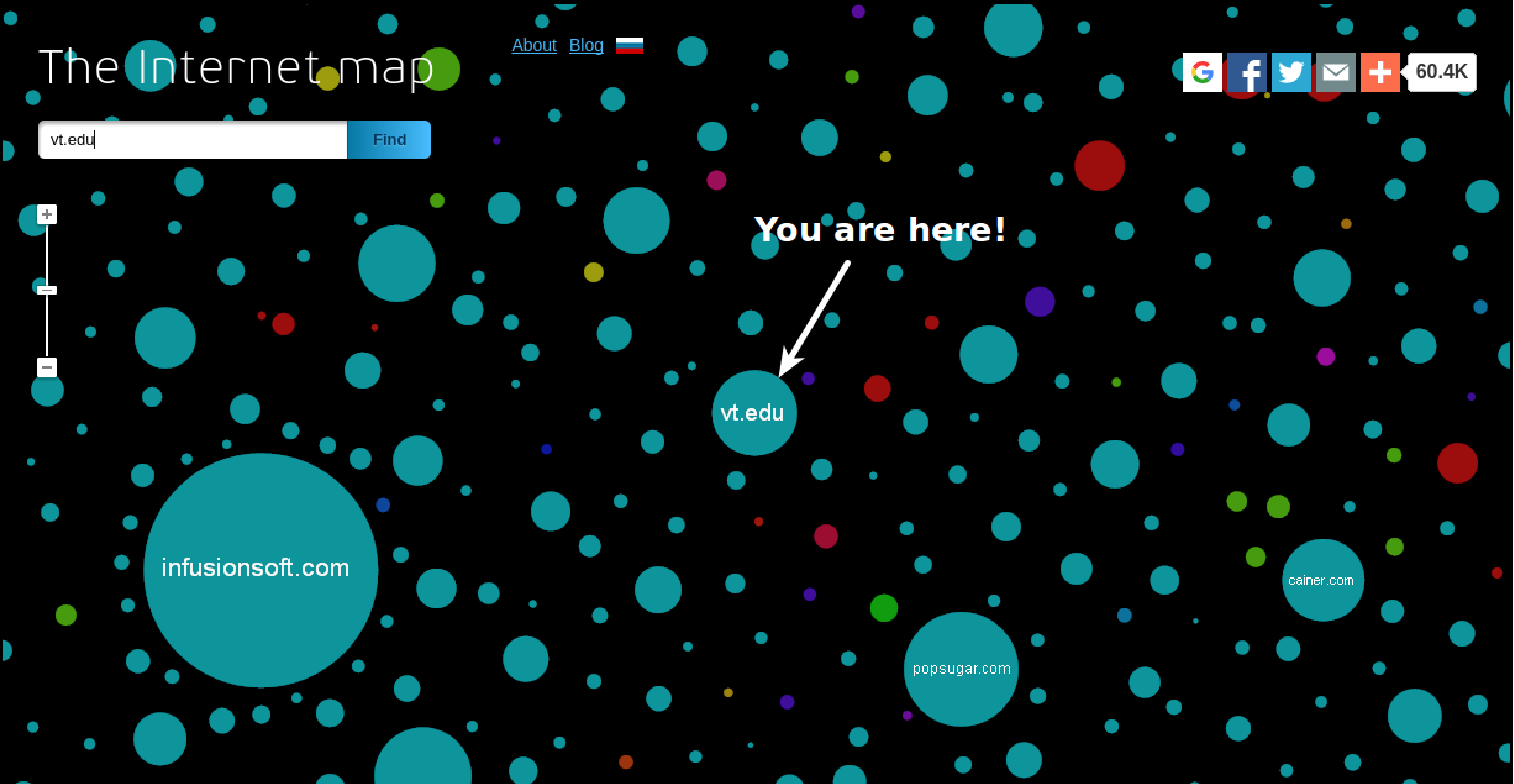


Image source: internet-map.net

2. The amount of supervision in a learning task depends on both the information content and noise level in the labels

3. Unsupervised learning is used by the brain

Webly Supervised Learning of Convolutional Networks

Xinlei Chen, Abhinav Gupta

ICCV, 2015

The Problem

- Investigation of webly-supervised learning of CNNs
- Can CNNs be trained for easy categories using images retrieved by search engines and from the web at large?
- How well do webly-supervised CNNs generalize to vision tasks?
- Main contributions:
  - Trained a webly-supervised CNN on PASCAL VOC 2012 which outperforms ImageNet
  - Webly-supervised learning works for object localization and for training R-CNN style detectors
  - State-of-the-art performance on Pascal VOC data without training on VOC dataset
  - Competitive performance on scene classification

Motivation

- Supervised learning using CNNs has achieved state-of-the-art success in a variety of vision tasks
- How to improve performance?
  - Deeper networks are better, especially with more data
  - But human labled data is expensive and innefficient
  - Web data is biased and noisy but is nearly infinite in scale (and continuously growing)
  - Exploit web data to train CNNs
  - Okay, but how?

Google vs Flickr

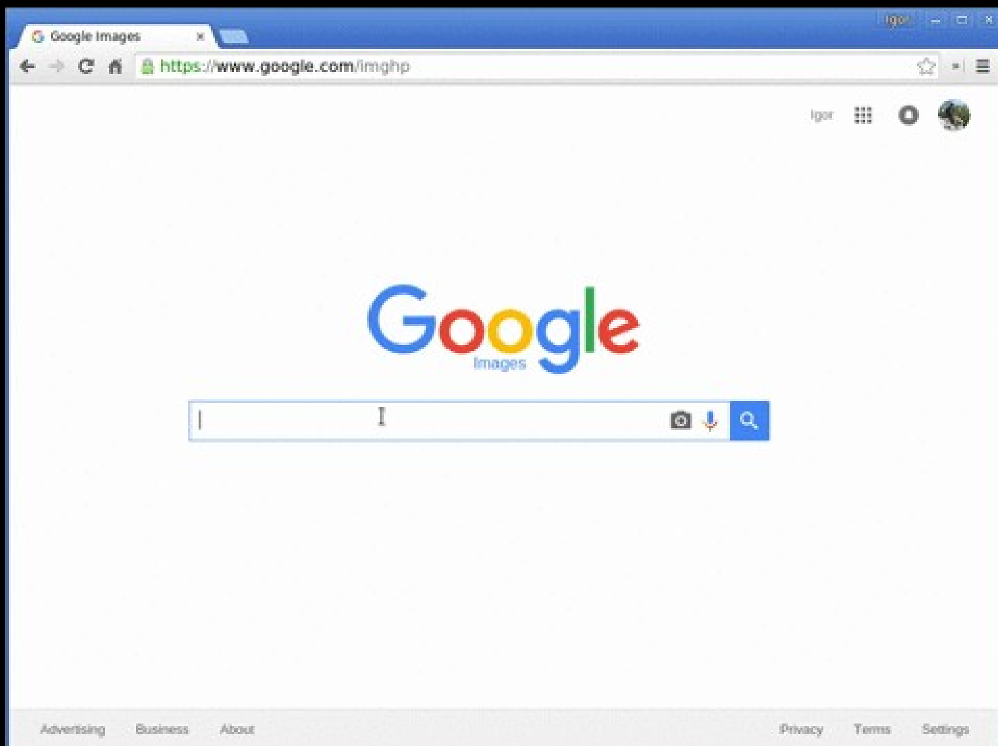


Image source: google.com

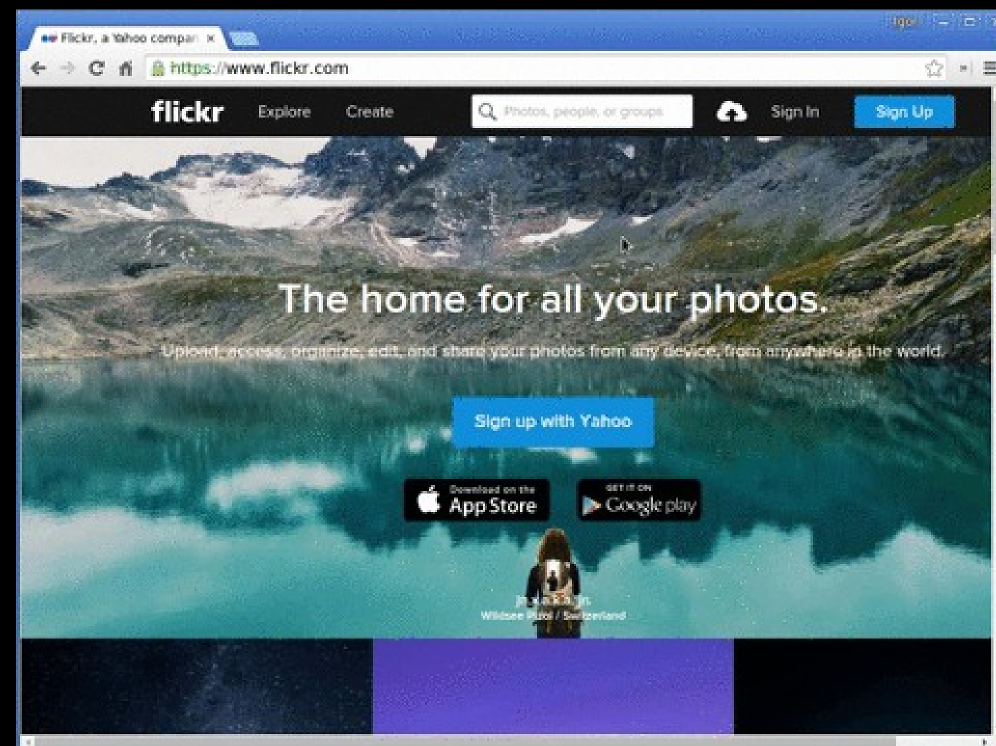


Image source: flickr.com

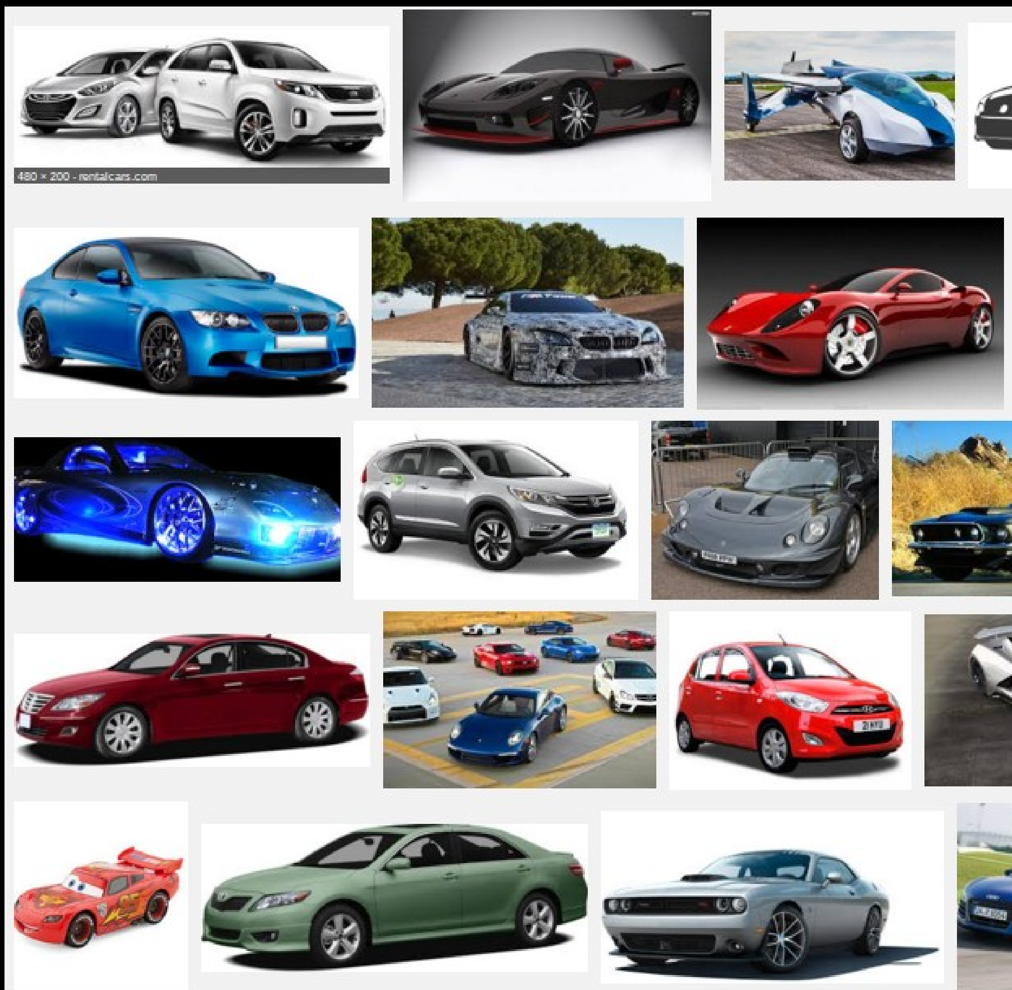


Image source: google.com

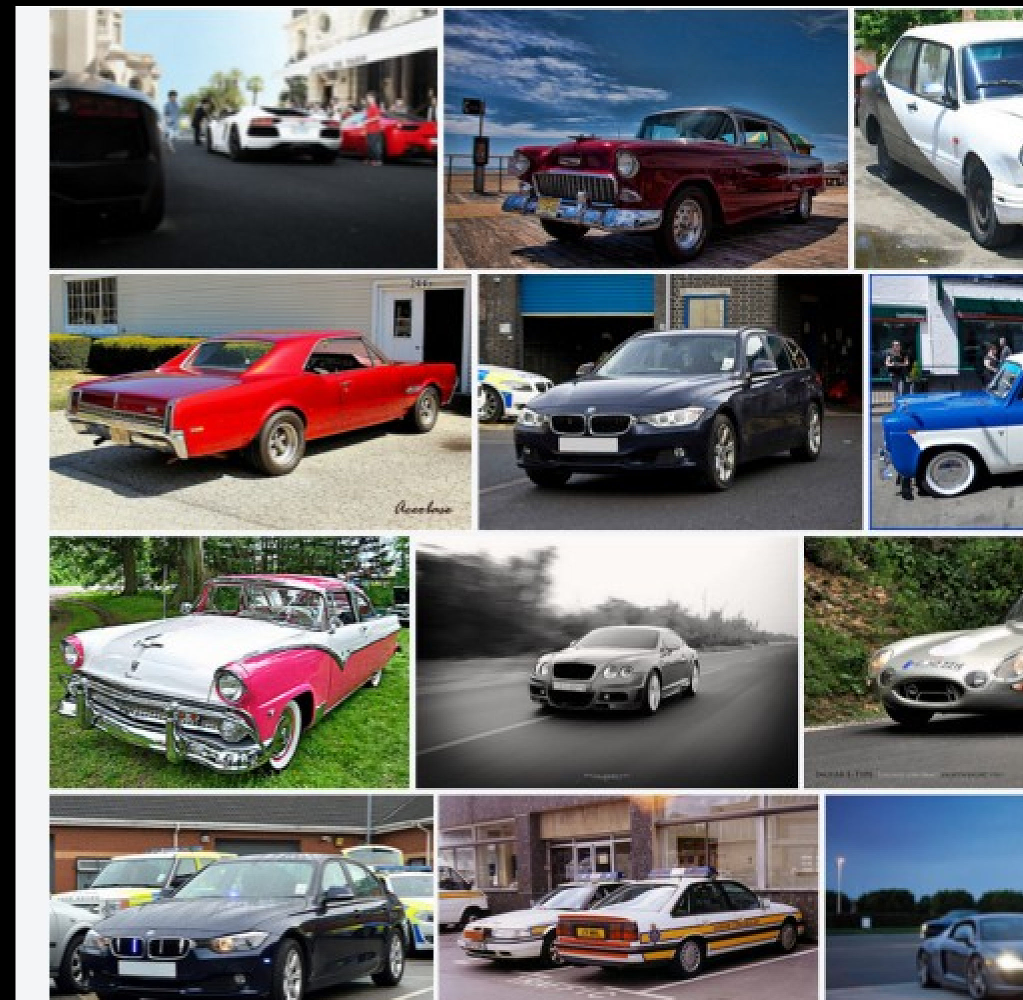


Image source: flickr.com

Image source: google.com

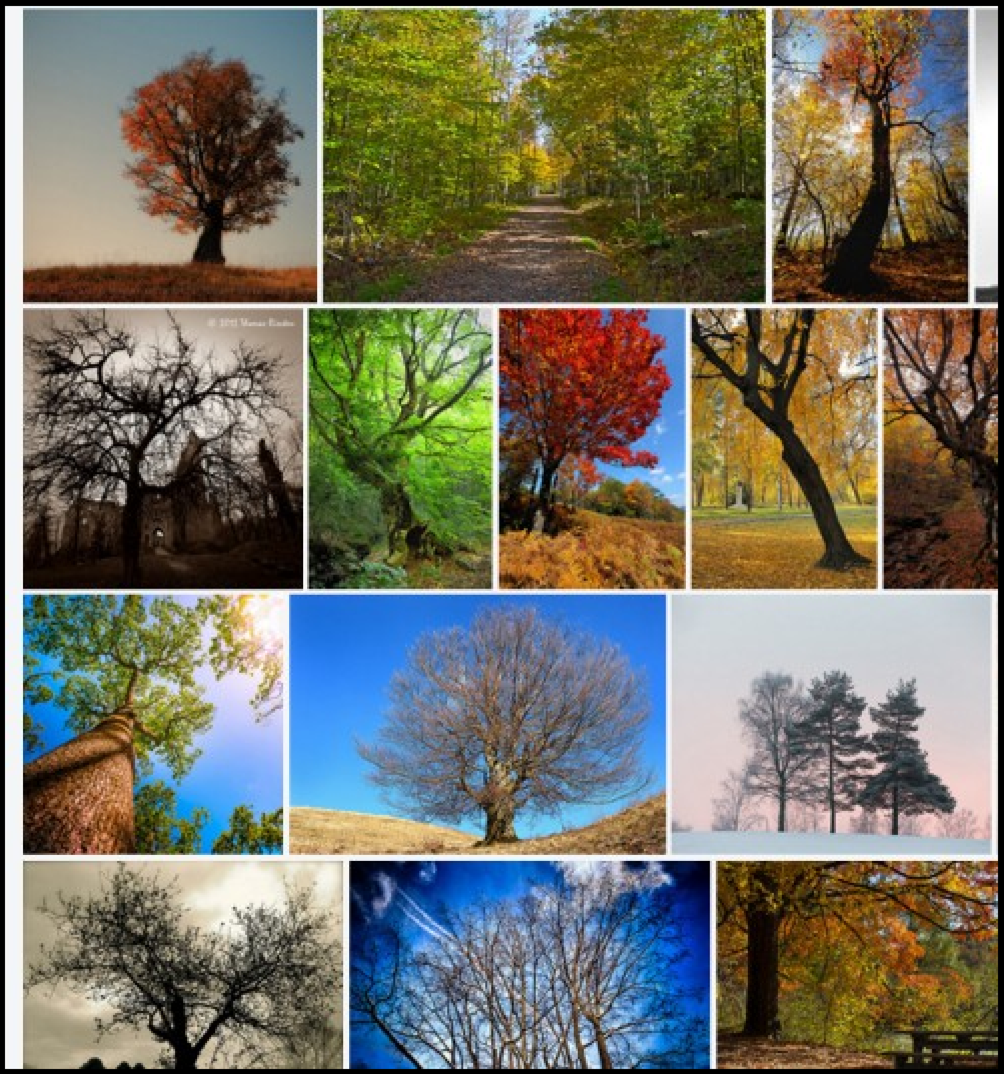


Image source: flickr.com

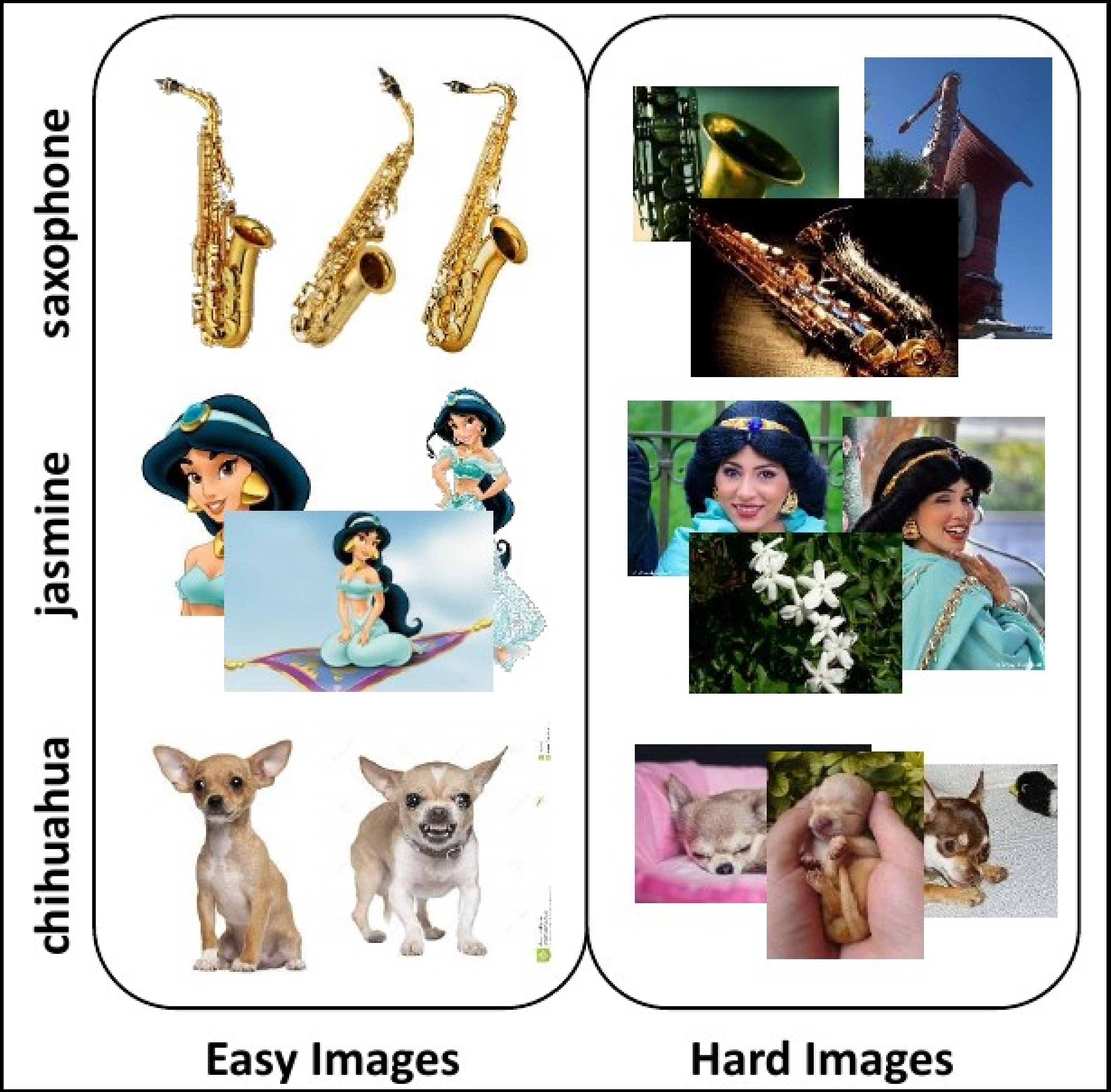


Image source: reference paper

Bootstrapping

- The goal is to learn a model on Flickr-like images, but these images often have very noisy tags
- Bootstrap CNN training with easy, noise-free examples first and then follow with a more comprehsensive learning proceedure

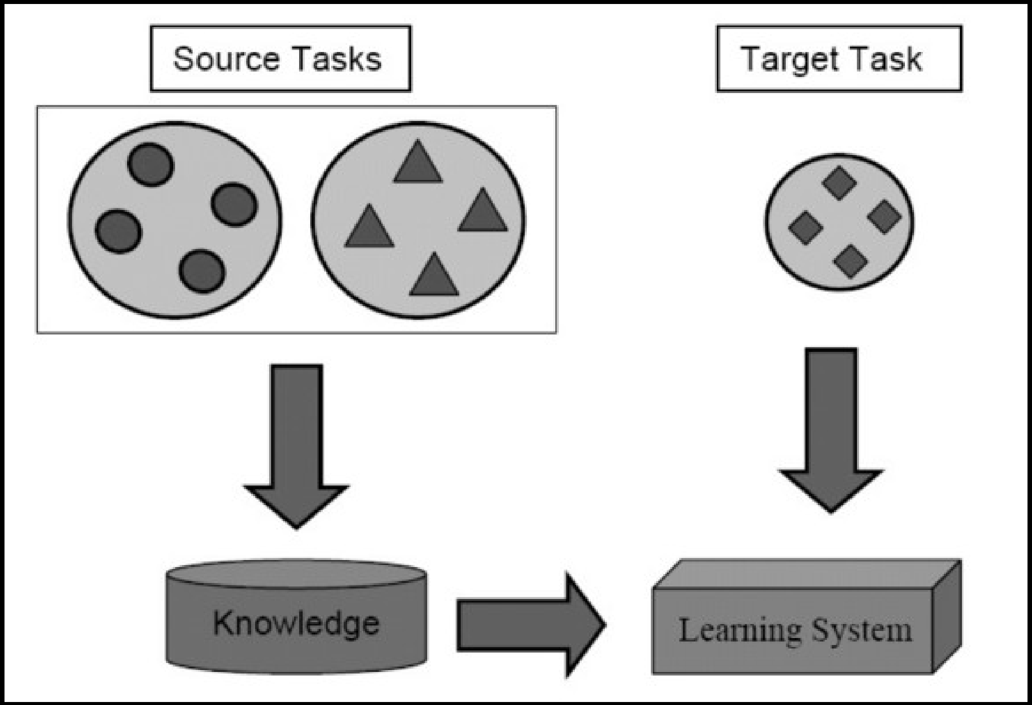


Image source: Sinno Jialin Pan, Qiang Yang, "A Survey on Transfer Learning”

Approach

- Could naively train a CNN on random image/tag pairs from the web
- Instead, first train the CNN model from scratch using easy images downloaded from Google search queries
- Then finetune this representation using harder Flickr images under specific constraints determined by a relationship graph
- Use the confusion matrix from the initial training done on easy images as the relationships between labels

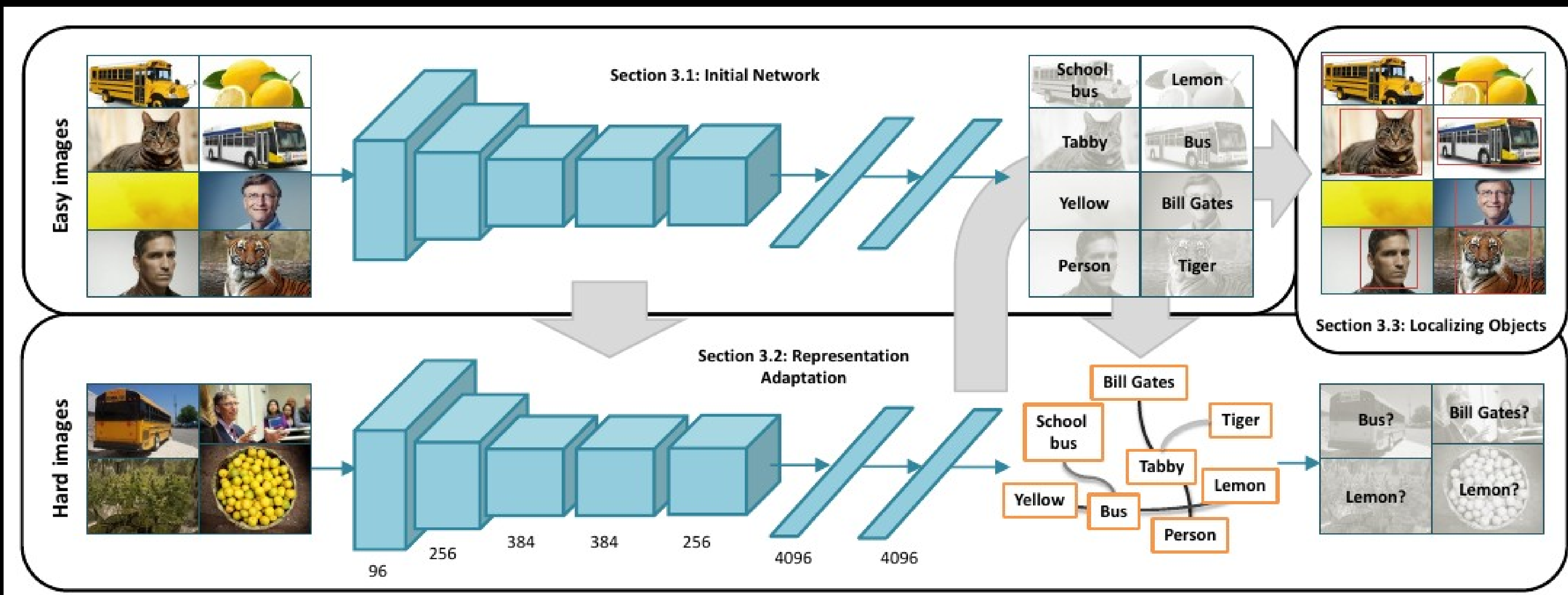


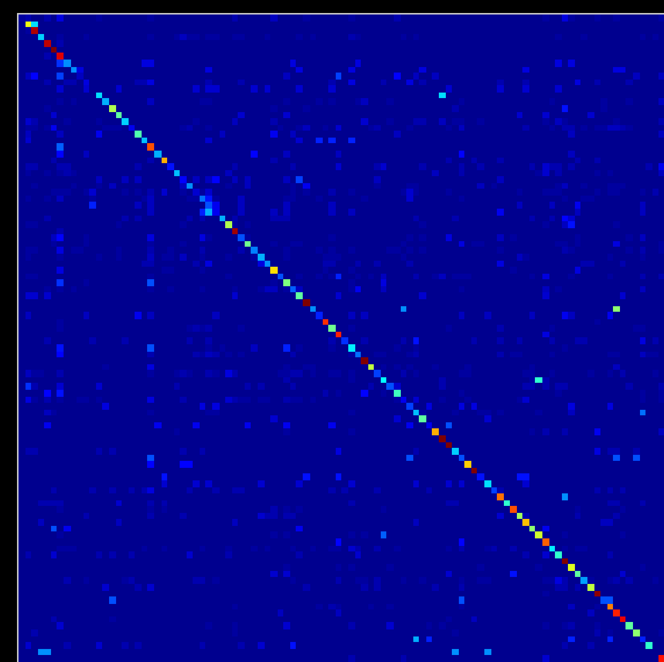
Image source: reference paper

Confusion Matrix

- Suppose a classifier was trained to distinguish between cats, dogs, and rabbits
- Dataset contains 27 images of 8 cats, 6 dogs, and 13 rabbits
- Diagonals of confusion matrix are all of the correct guesses

		Predicted		
		Cat	**Dog**	**Rabbit**
Actual class	**Cat**	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

Example source: wikipedia



Example source: Andrea Vedaldi

Approach

- Three lists of categories are constructed from the ImageNet Challenge, SUN database, and NEIL knowledge base
- These category names are used as Google search queries to construct a dataset of ~600 images/query
- BVLC reference network is trained on this dataset in a supervised manner using Caffe
- After convergence, the network has learned good low-level filters but is biased towards simple images
- Now construct a dataset of more realistic Flickr images found using the same search queries

Relationship Graph

- Object categories have various complex relationships such as hierarchies, dependencies, similarities, restrictions, etc., all of which together form an ontology representable as a graph
- Approach taken in the paper is to simply use the confusion matrix as the relationships
- For any pair of concepts *i* and *j*, the relationship R_{ij} is defined as

$$R_{ij} = P(i|j) = \frac{\sum_{k \in C_i} CNN(j|I_k)}{|C_i|},$$

  where C_i is the index set for images that belong to concept *i*, and given pixel values I_k, $CNN(j \mid I_k)$ is the network's belief on how likely image *k* belongs to concept *j*
- Choose only the top $K = 5$

Relationship Graph

- Relationship graph is a way to characterize the label-flip noise
- For a class label l_k, softmax loss is

$$L = \sum_k \sum_i R_{il_k} \log(CNN(i|I_k)),$$

- Relationship matrix R is kept fixed after being learned using the initial network

Object Localization

- Need to clean web data and localize objects to train a R-CNN detector
- But CNN only distinguishes small set of classes and is spatially invariant
- Google images have centered-biased images so they are used as bounding box seeds
- Nearest Neighbor propogation to find neighbors and EdgeBox to find candidate windows
- Agglomerative clustering merges NN sets bottom up to form subcategories and R-CNN detector is trained on each category using all clustered bounding boxes
- Random patches from YFCC are used as negatives
- Positive bounding boxes are increased using EdgeBox and by using the relationship graph to expand the category
- Final SVM is trained

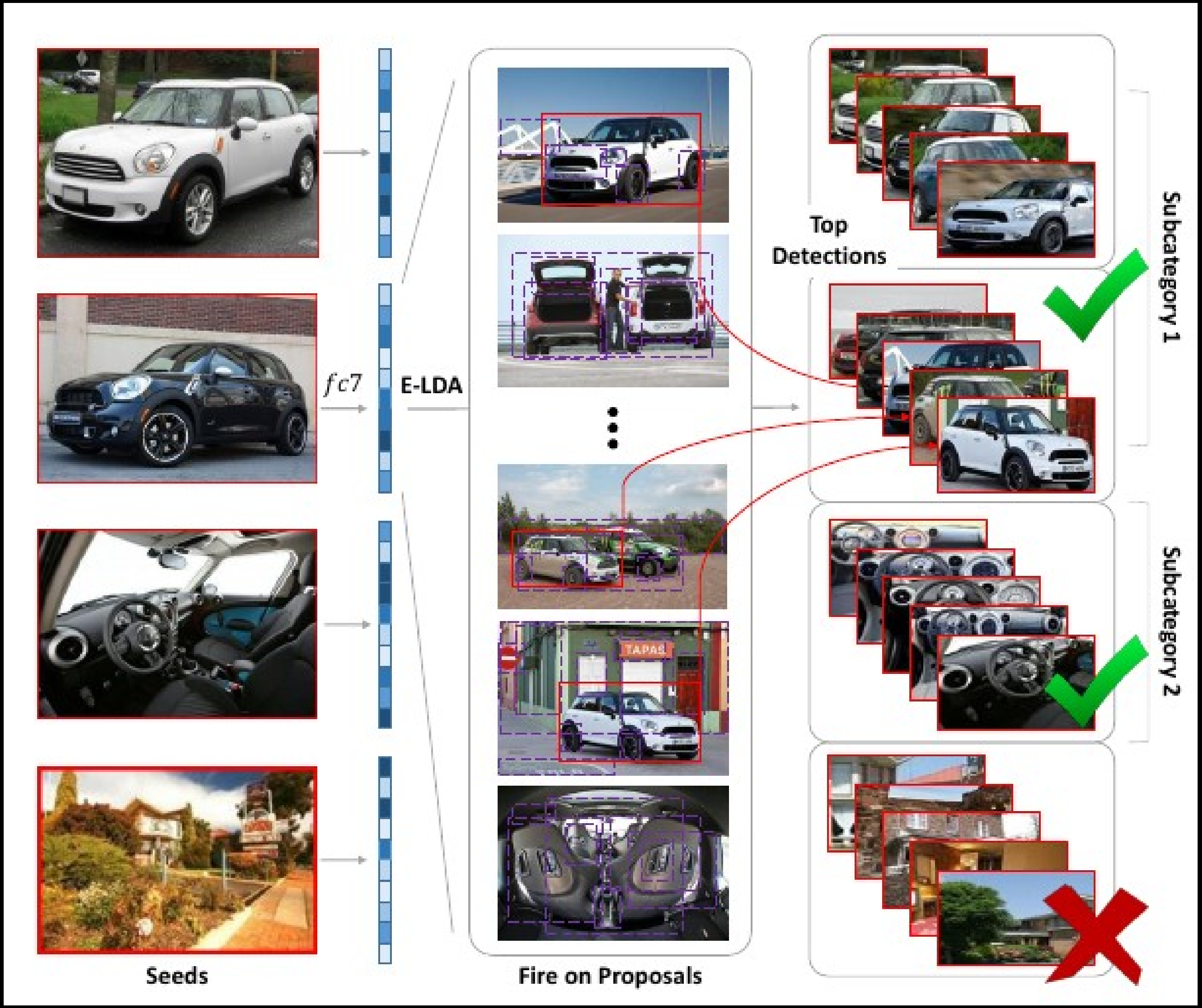


Image source: reference paper

Image source: reference paper

Implementation Details

- Networks are trained in Caffe
  - Batch size is 256
  - Learning rate starts at 0.01 and reduced by a factor of 10 every 150k iterations
  - Training stops after 450k iterations
- 2,240 objects, 89 attributes, 874 scenes
- GoogleO (object-attribute network):
  - ~1.5 million images from Google image search
  - Later fine-tuned with ~1.2 million Flickr images with relationship graph regularization (FlickrG) and without (FlickrF) for 100k iterations and step size of 30k
  - Baselines: CNN learned using Flickr images alone (FlickrS) and combined Google and Flickr images (GFAll)
- GoogleA (all-included network):
  - ~2.1 million images from Google image search (add scene images)

Visualizing Confusion Matrix

- Diagonal of confusion matrix is ranked in descending order and 3 random categories are sampled from top, bottom, and middle of list

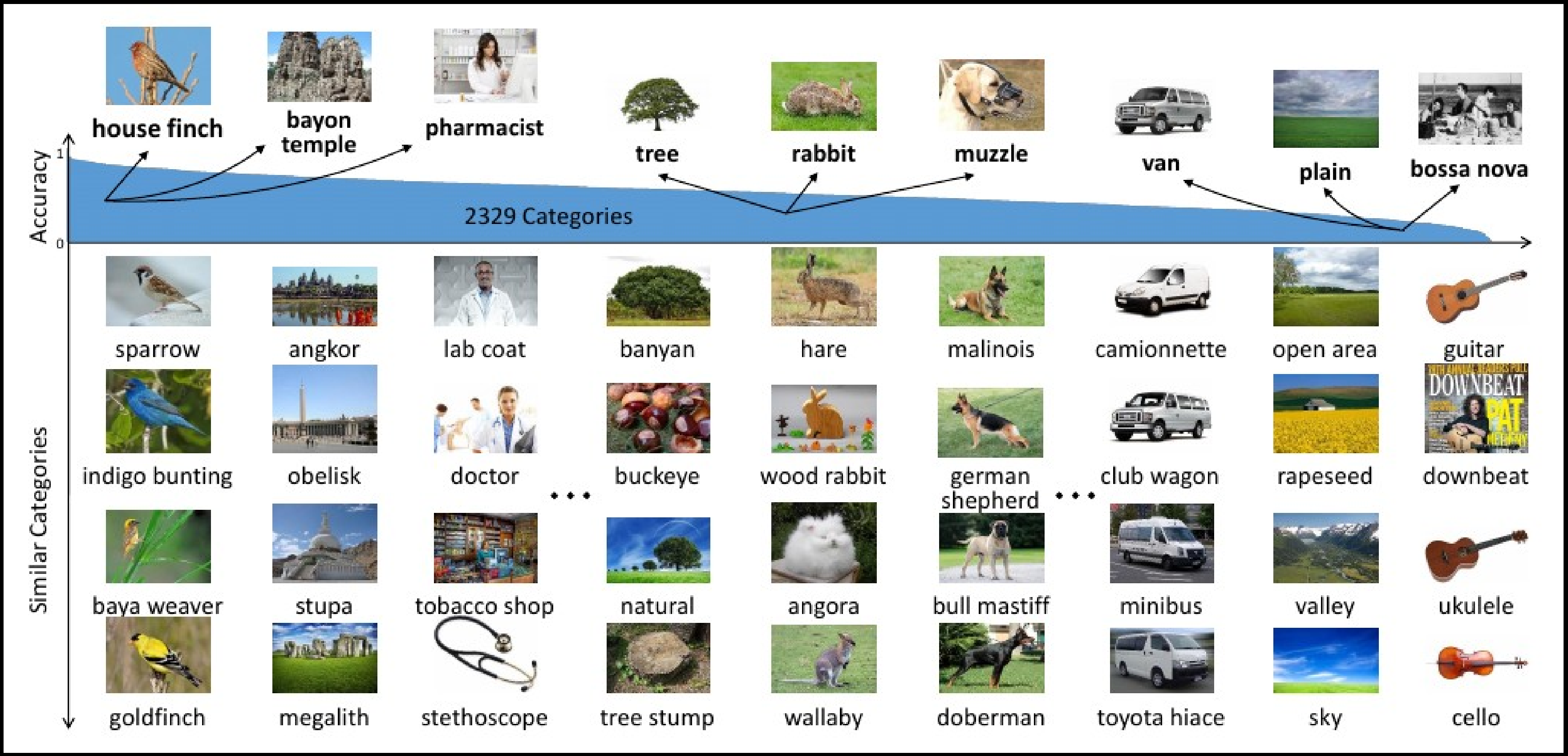


Image source: reference paper

Object Detection (PASCAL VOC 2007)

	VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	pers	plant	sheep	sofa	train	tv	mAP
w/o VOC FT	ImageNet [20]	**57.6**	57.9	38.5	**31.8**	23.7	51.2	58.9	**51.4**	20.0	**50.5**	**40.9**	**46.0**	**51.6**	55.9	**43.3**	23.3	**48.1**	35.3	**51.0**	**57.4**	**44.7**
	GoogleO [Obj.]	57.1	59.9	35.4	30.5	21.9	53.9	59.5	40.7	18.6	43.3	37.5	41.9	49.6	57.7	38.4	22.8	45.2	37.1	48.0	54.5	42.7
	GoogleA [Obj. + Sce.]	54.9	58.2	35.7	30.7	22.0	54.5	59.9	44.7	19.9	41.0	34.5	40.1	46.8	56.2	40.0	22.2	45.8	36.3	47.5	54.2	42.3
	FlickrS [Flickr Obj.]	50.0	55.9	29.6	26.8	18.7	47.6	56.3	34.4	14.5	35.9	33.3	34.2	43.2	52.2	36.7	21.5	43.3	31.6	48.5	48.4	38.1
	GFAll [All Obj., 1-stage]	52.1	57.8	38.1	25.6	21.2	47.6	56.4	43.8	19.6	42.6	30.3	37.6	45.1	50.8	39.3	22.9	43.5	34.2	48.3	52.2	40.5
	FlickrF [2-stage]	53.9	60.7	37.0	31.6	23.8	**57.7**	60.8	44.1	20.3	46.5	31.5	39.8	49.7	59.0	41.6	23.0	44.4	36.2	49.9	56.2	43.4
	FlickrG [2-stage, Graph]	55.3	**61.9**	**39.1**	29.5	**24.8**	55.1	**62.7**	43.5	**22.7**	49.3	36.6	42.7	48.9	**59.7**	41.2	**25.4**	47.7	**41.9**	48.8	56.8	**44.7**
w/ VOC FT	VOC-Scratch [2]	49.9	60.6	24.7	23.7	20.3	52.5	64.8	32.9	20.4	43.5	34.2	29.9	49.0	60.4	47.5	28.0	42.3	28.6	51.2	50.0	40.7
	ImageNet [20]	64.2	**69.7**	**50.0**	**41.9**	**32.0**	62.6	71.0	**60.7**	**32.7**	58.5	46.5	**56.1**	60.6	66.8	54.2	31.5	52.8	48.9	57.9	**64.7**	**54.2**
	GoogleO	**65.0**	68.1	45.2	37.0	29.6	65.4	73.8	54.0	30.4	57.8	48.7	51.9	**64.1**	64.7	54.0	**32.0**	**54.9**	44.5	57.0	64.0	53.1
	GoogleA	64.2	68.3	42.7	38.7	26.5	65.1	72.4	50.7	28.5	**60.9**	**48.8**	51.2	60.2	65.5	**54.5**	31.1	50.5	**48.5**	56.3	60.3	52.3
	FlickrG	63.7	68.5	46.2	36.4	30.2	**68.4**	**73.9**	56.9	31.4	59.1	46.7	52.4	61.5	**69.2**	53.6	31.6	53.8	44.5	**58.1**	59.6	53.3

Object Detection (PASCAL VOC 2012)

	VOC 2012 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
w/ VOC FT	ImageNet [20]	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	**66.2**	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6	49.6
	ImageNet-TV	**73.3**	67.1	46.3	31.7	30.6	59.4	61.0	**67.9**	27.3	**53.1**	39.1	64.1	**60.5**	70.9	57.2	26.1	**59.0**	40.1	56.2	**54.9**	52.3
	GoogleO	72.2	67.3	46.0	**32.3**	**31.6**	**62.6**	62.5	66.5	27.3	52.1	38.9	64.0	59.1	71.6	58.0	27.2	57.6	41.3	56.3	53.7	52.4
	FlickrG	72.7	**68.2**	**47.3**	32.2	30.6	62.3	**62.6**	65.9	**28.1**	52.2	**39.5**	65.1	60.0	**71.7**	**58.2**	**27.3**	58.0	**41.5**	**57.2**	53.8	**52.7**

Object Localization (PASCAL VOC 2007)

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	**mAP**
LEVAN [14]	14.0	36.2	12.5	10.3	9.2	35.0	**35.9**	8.4	**10.0**	17.5	6.5	12.9	**30.6**	27.5	6.0	1.5	18.8	10.3	23.5	16.4	17.1
GoogleO	30.2	34.3	16.7	13.3	6.1	43.6	27.4	22.6	6.9	16.4	10.0	21.3	25.0	35.9	7.6	9.3	21.8	17.3	31.0	18.1	20.7
GoogleA	29.5	38.3	15.1	14.0	9.1	44.3	29.3	24.9	6.9	15.8	9.7	22.6	23.5	34.3	9.7	12.7	21.4	15.8	33.4	19.4	21.5
FlickrG	32.6	42.8	19.3	13.9	9.2	46.6	29.6	20.6	6.8	17.8	10.2	22.4	26.7	40.8	11.7	**14.0**	19.0	19.0	34.0	21.9	22.9
FlickrG-EA	**32.7**	**44.3**	17.9	14.0	**9.3**	**47.1**	26.6	19.2	8.2	18.3	10.0	22.7	25.0	42.5	12.0	12.7	**22.2**	20.9	35.6	18.2	23.0
FlickrG-CE	30.2	41.3	**21.7**	**18.3**	9.2	44.3	32.2	**25.5**	9.8	**21.5**	**10.4**	**26.7**	27.3	**42.8**	**12.6**	13.3	20.4	**20.9**	**36.2**	**22.8**	**24.4**

Failure Modes

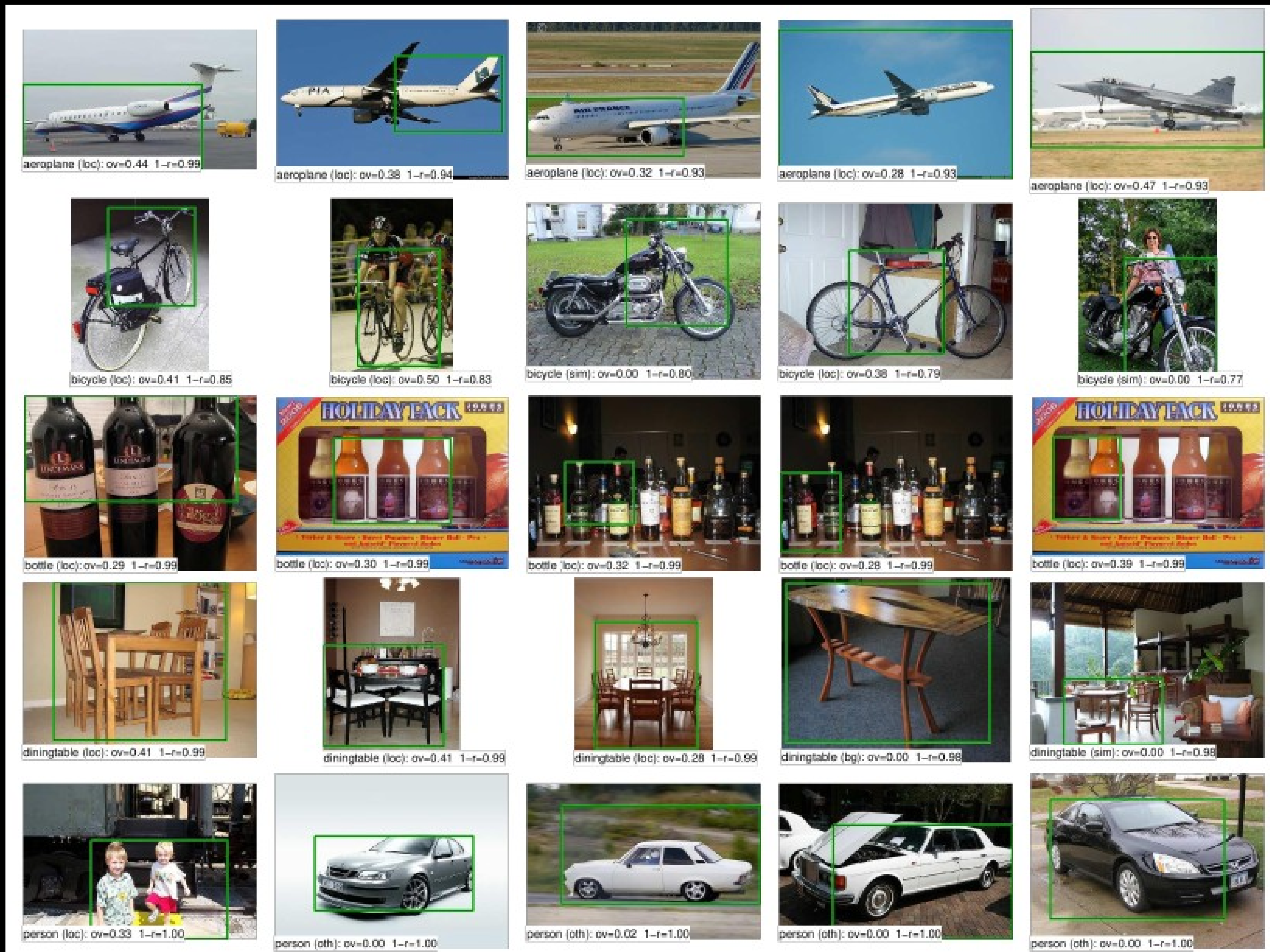


Image source: reference paper

Failure Modes

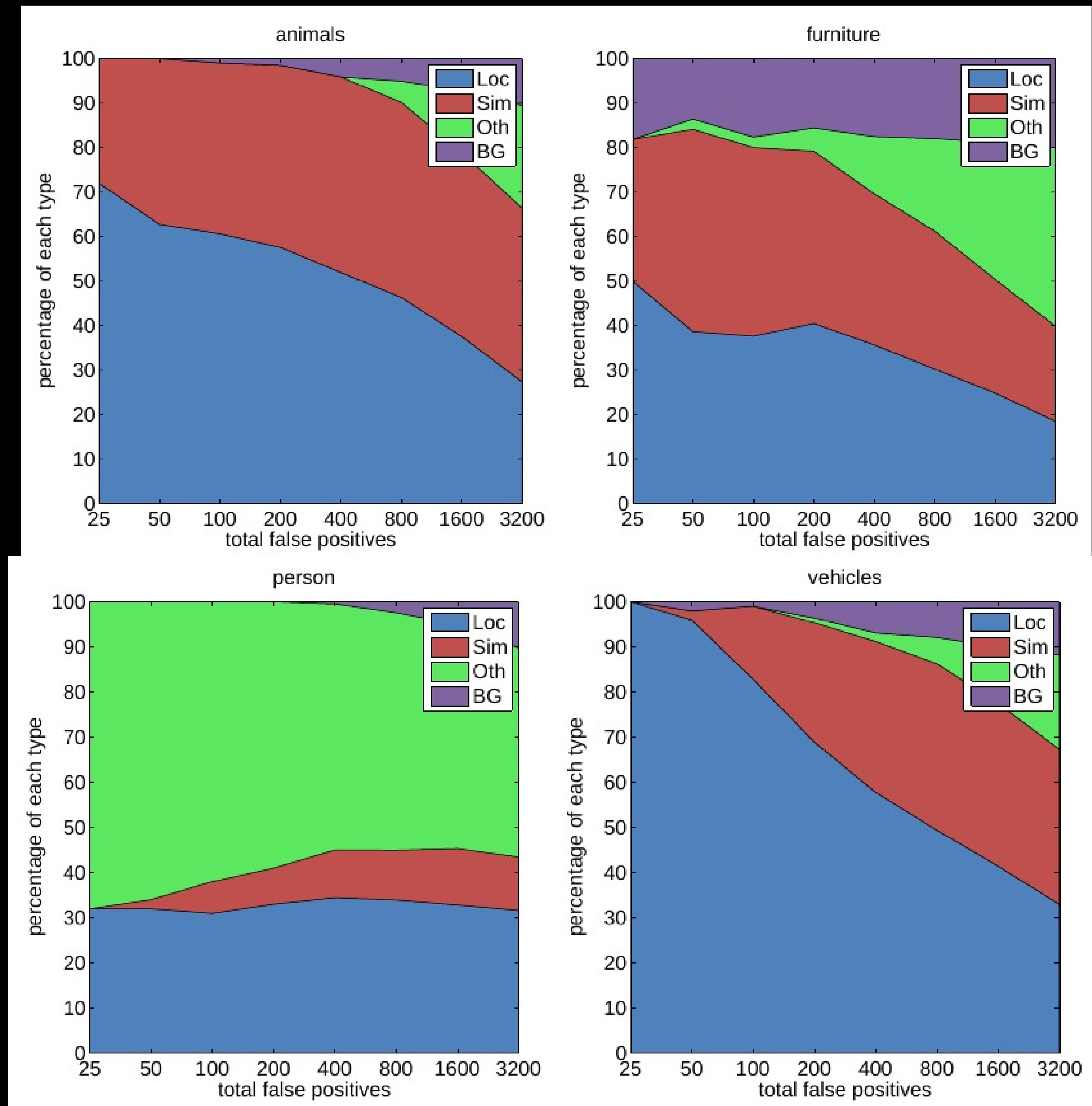


Image source: reference paper

Scene Classification (MIT-67)

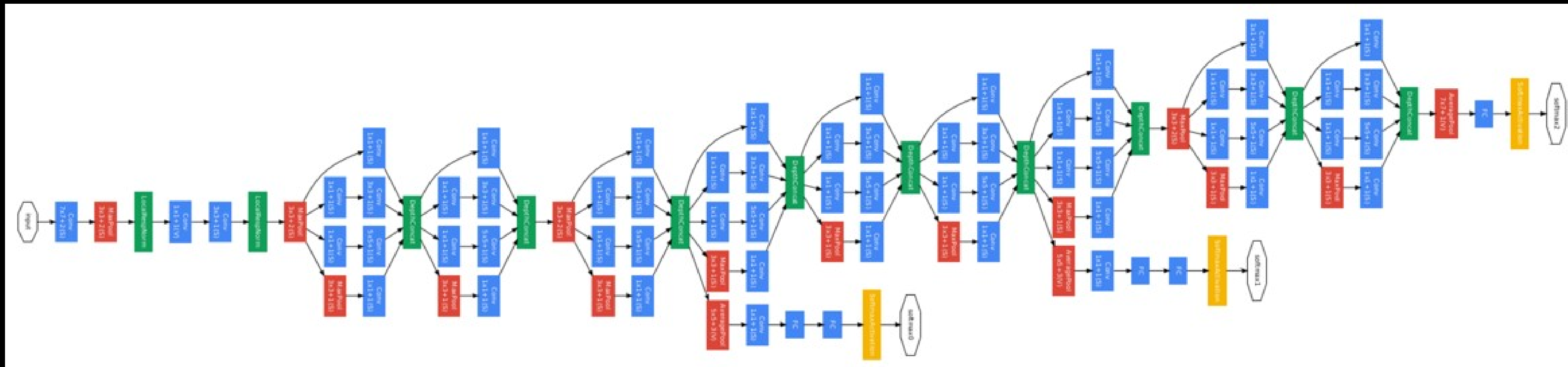
Indoor-67	Accuracy
ImageNet [62]	56.8
OverFeat [42]	58.4
GoogleO [Obj.]	58.1
FlickrG [Obj.]	59.2
GoogleA [Obj. + Sce.]	**66.5**

Conclusion

- Two-stage bootstrapping approach to training CNNs using web data
- First train on easy images downloaded from Google which is used to initialize the network and the relationship graph
- Then finetune the network on Flickr images and use the relationship graph for regularization
- Network has similar results to ImageNet pre-trained CNN on VOC 2007 and outperforms on VOC 2012 for object detection
- Object localization and scene classification can be done webly-supervised

LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, Jianxiong Xiao
arXiv, 2015

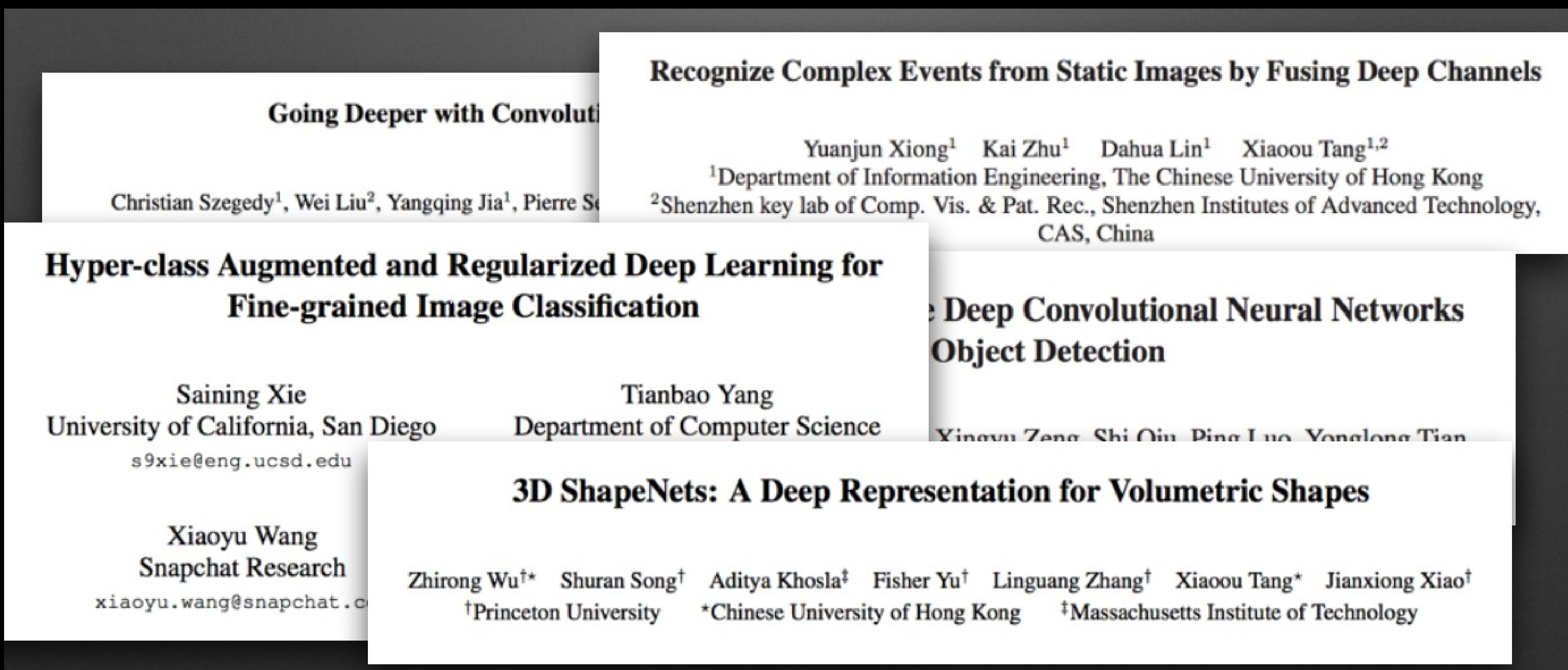


Models

Data

Motivation

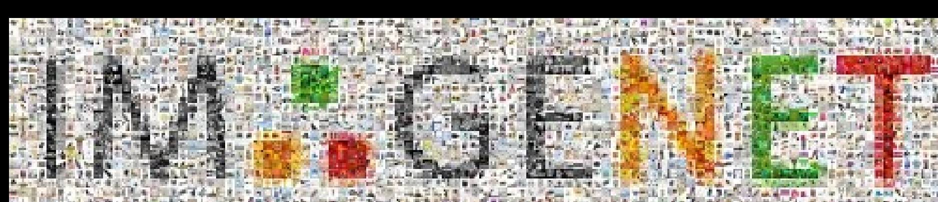
- Data hungry algorithms everywhere
- ImageNet is out-of-date
- 59 papers with “Deep” in the title in CVPR 2015



Slide credit: Fisher Yu

Problem Statement

- Goal is to build a dataset containing hundreds of millions of images
- Why?
  - Deep learning models usually have millions of parameters
  - Searching for the optimal settings requires a massive amount of training data with accurate labels
  - Human labeled data is expensive, inneficient, and contains mistakes
  - Partially automate data collection using deep learning methods and statistical guarantees to catch up with the progress of algorithms and computers



x 100

Image credit: Andrej Karpathy

Gathering Images

- LSUN dataset has the same scene categories as SUN
- Images are acquired using Google search queries combined with 696 manually chosen common adjectives
- Around 600 million images downloaded
- Duplicate images are ignored (not removed)

Approach

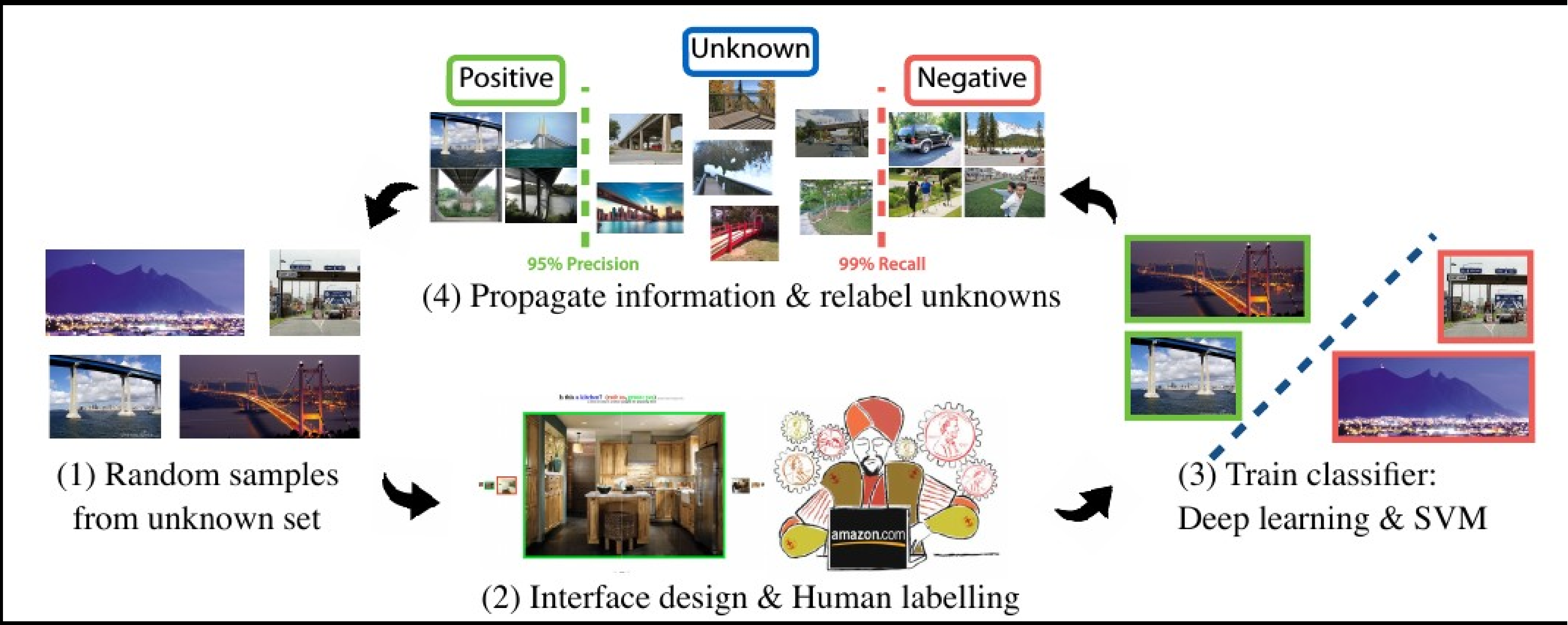
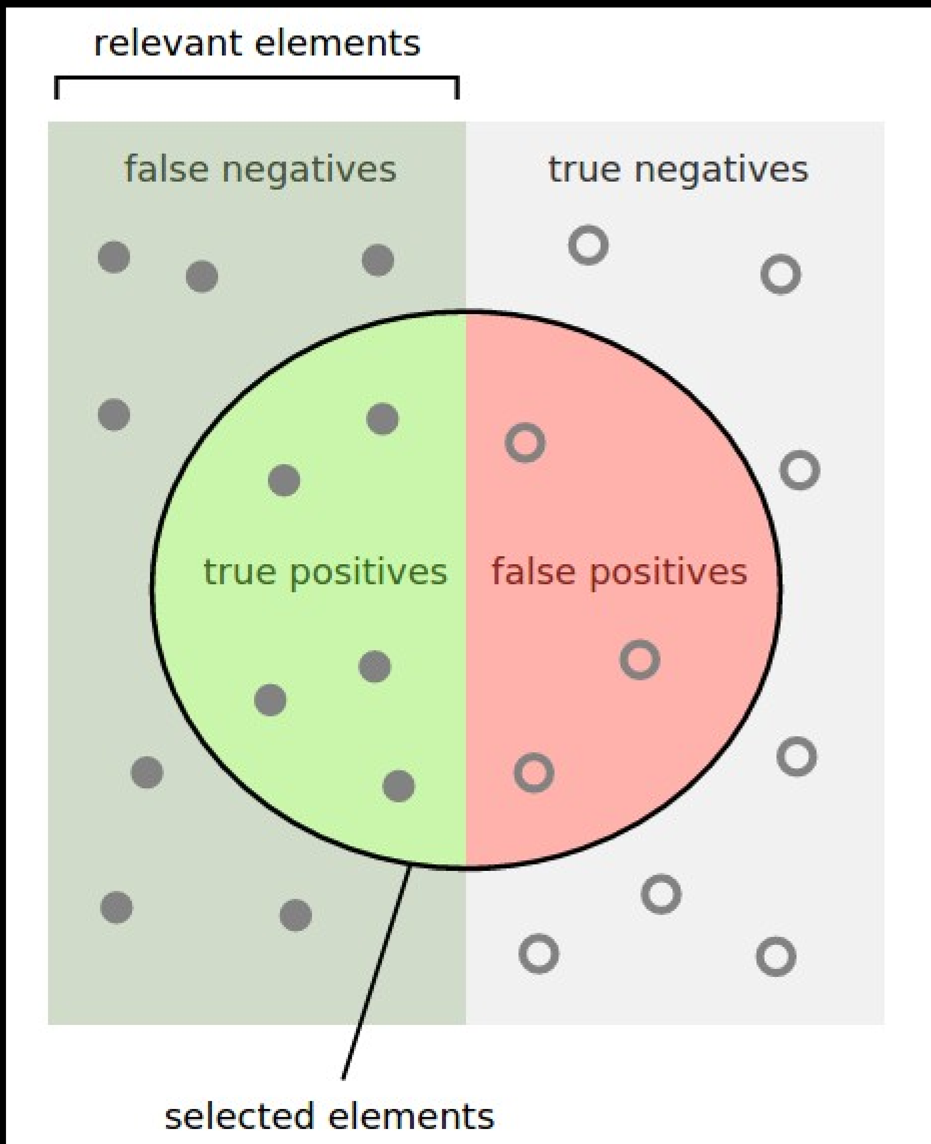


Image source: reference paper

Precision and Recall



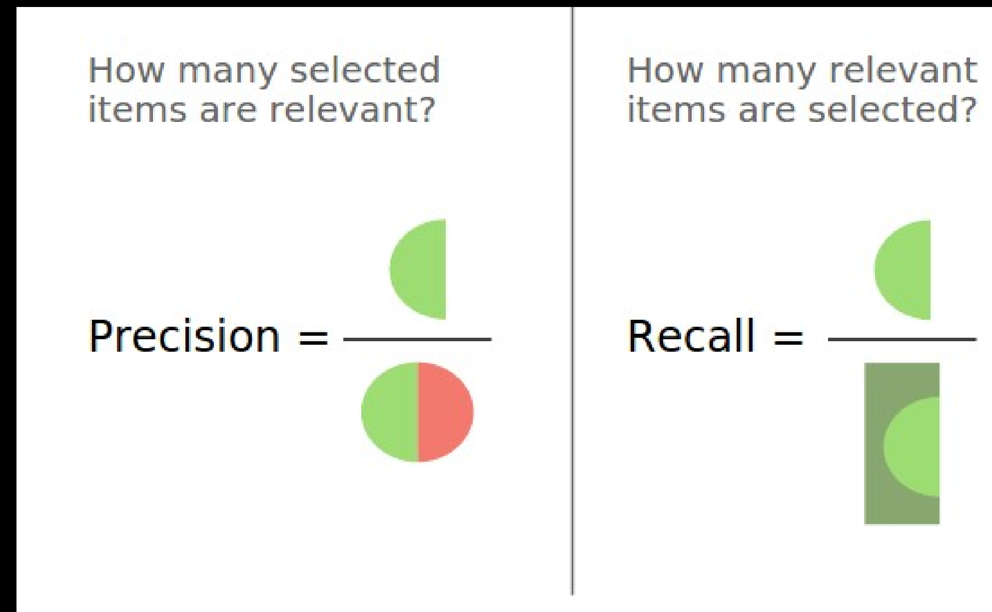


Image source: Wikipedia

Example: Train an object detector to detect cats in a scene containing 9 dogs and some cats. 4 dogs are correctly detected while 3 of the detected dogs are actually cats. P = 4/7 while R = 4/9.

AMT Labeling Interface

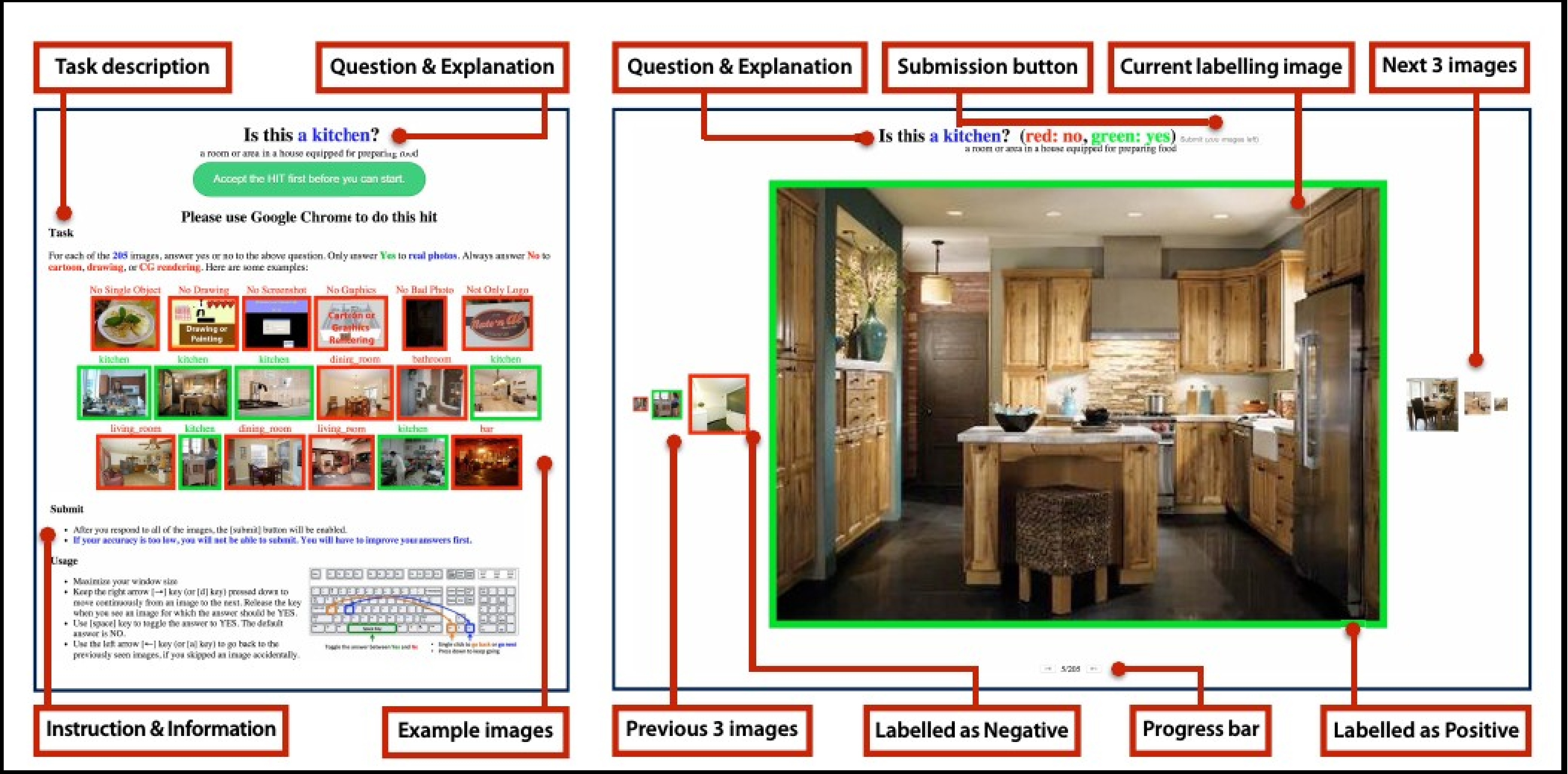


Image source: reference paper

Lifetime of AMT Hit

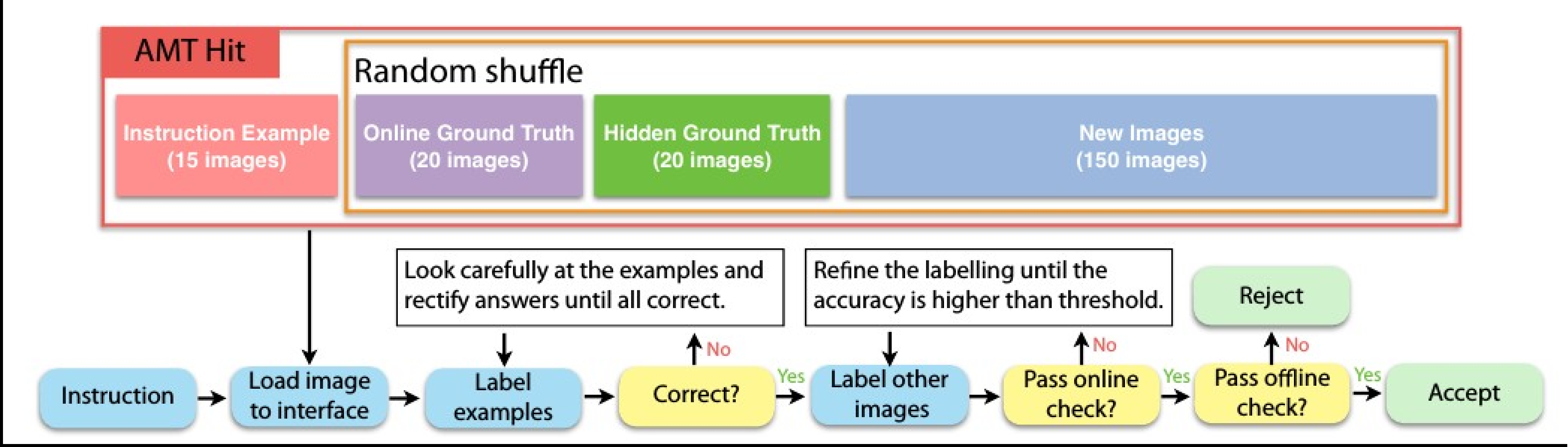
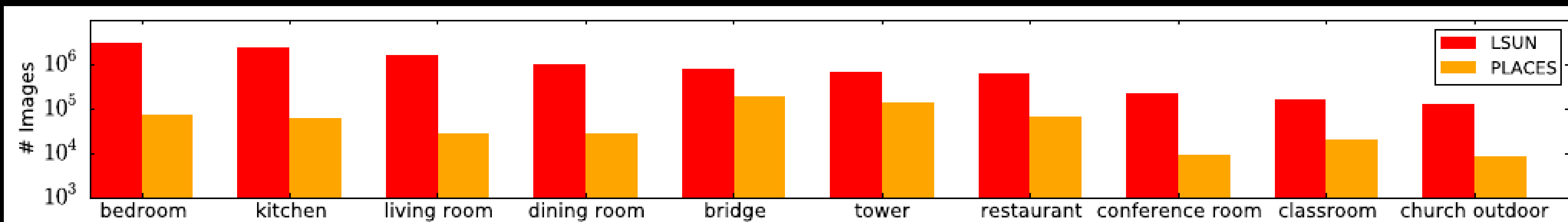


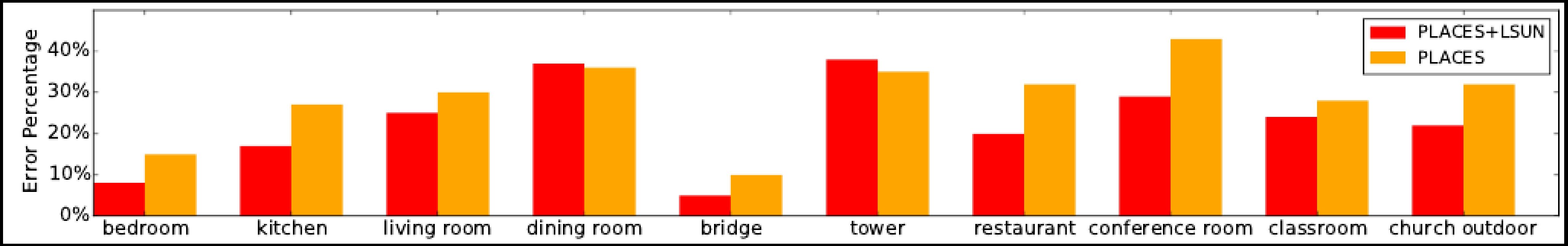
Image source: reference paper



Images
10^6
10^5
10^4
10^3
LSUN
PLACES
bedroom
kitchen
living room
dining room
bridge
tower
restaurant
conference room
classroom
church outdoor

Iteration	Method	Positive images	Precision	Positive labels	Label Ratio
0	Clustering	941,981	96.9%	9,515	1%
1	ConvNet Feature + SVM	1,918,913	96.4%	41,785	2.1%
2	ConvNet Feature + SVM	2,011,332	96.2%	88,259	4.4%
3	ConvNet Fine Tuning	2,140,763	96.1%	88,259	4.1%
4	ConvNet Fine Tuning	2,215,244	95.9%	147,453	6.6%

Experiments



Conclusion

- Datasets are a major roadblock to advancing progress visual recognition
- A large dataset called LSUN was created with millions of accurately labled images
- Simple experiments were demonstrated to show the datasets potential
- Construction of the dataset is still underway