Leveraging Human Perception and Computer Vision Algorithms for Interactive Fine-Grained Visual Categorization

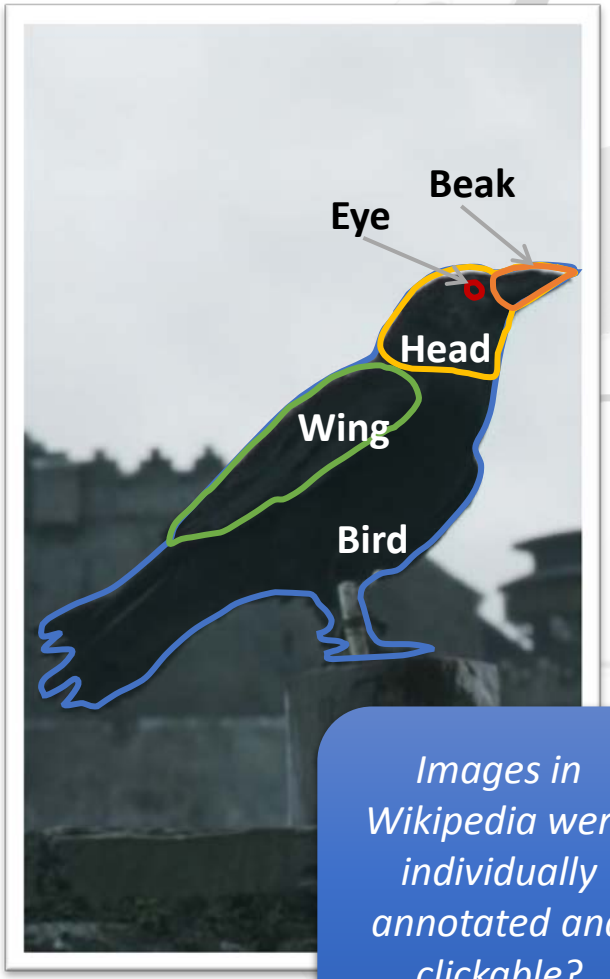
Catherine Wah

Computer Science and Engineering
University of California, San Diego

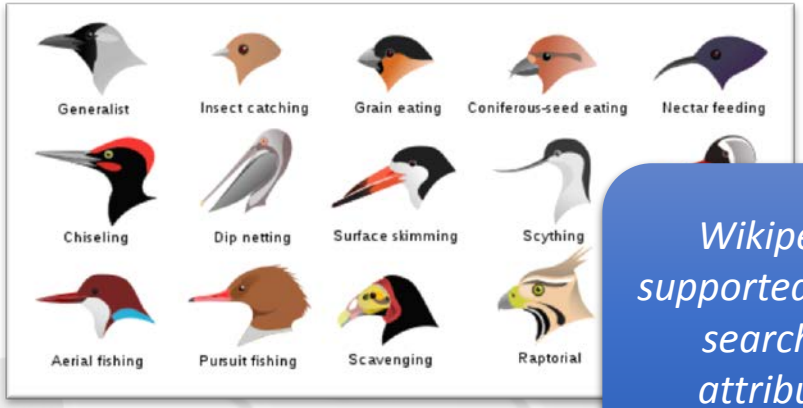
UCSD Jacobs | School of Engineering

OK Visipedia, what bird is this?
Common raven

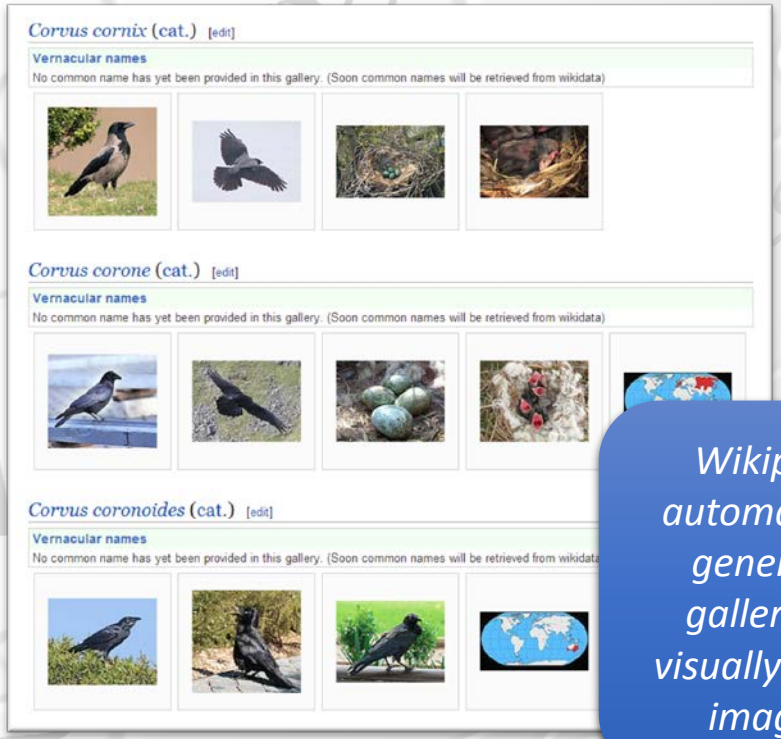
What if…



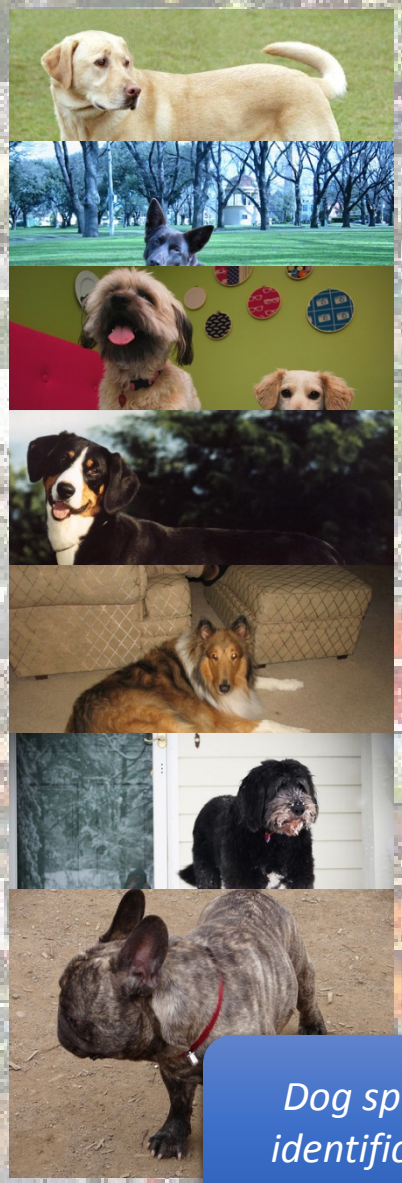
Images in Wikipedia were individually annotated and clickable?



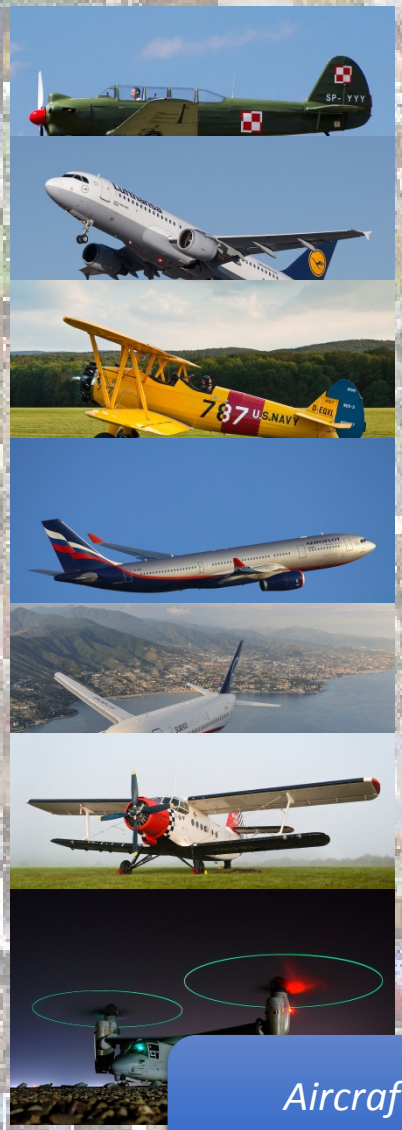
Wikipedia supported image search by attribute?



Wikipedia automatically generated galleries of visually similar images?



Dog species identification



Aircraft identification

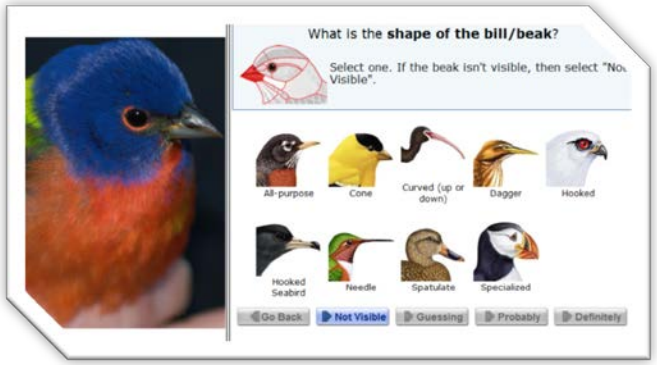
Vehicle make/ model classification

[Wikimedia Commons] [FGComp 2013]

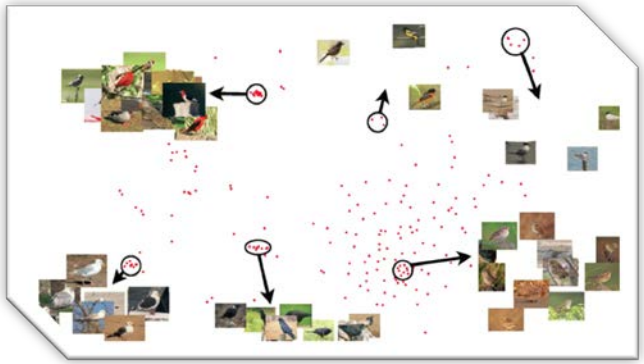
This talk

The motivating application for our work is **Visipedia**

We look at two paradigms for **interactive categorization**, based on:



Parts and Attributes



Similarity Learning

We focus on **fine-grained visual categories** (more on this later)

Visipedia

Visipedia

- User-generated knowledge base of visual objects
- Visual counterpart to Wikipedia
- Provide services like:
  - Better text-to-image search
  - Image-to-article visual search

Visipedia is built on
fine-grained visual categories.

[Perona 2010]

Fine-
Grained
Visual
Categories

Aircraft [Maji *et al*. 2013]

Birds [Wah *et al*. 2011]

Dogs [Khosla *et al*. 2011]

Fly larvae [Martínez-Muñoz *et al*. 2009]

Butterflies [Wang *et al*. 2009]

Plants [Belhumeur *et al*. 2008]

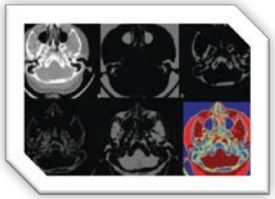
Flowers [Nilsback & Zisserman 2008]

…

Where do humans come in?

Humans are used for various tasks in computer vision applications

- Image annotation, debugging algorithms, interactive segmentation, active learning, *etc*.

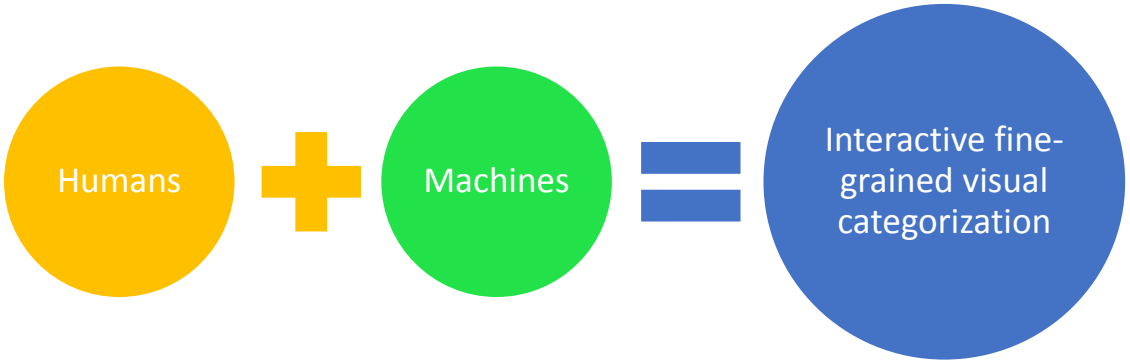


Expert systems

Active testing

Relevance feedback

We combine the strengths of humans and machines to perform interactive categorization.



[Geman & Jedynak 1993, Ferecatu & Geman 2009]

By your powers combined…

Standard classification methods fail due to:

- Insufficient training data
- High intraclass variation
- Low interclass variation

Humans can help boost performance until computer vision algorithms improve.

Strength	Machines	Humans
Recognizing fine-grained categories	No	No (non-experts)
Recognizing attributes	No	Yes
Judging visual similarity	No	Yes
Localizing deformable objects	No	Yes
Memory	Infinite	Limited
No subjective/perceptual bias	Yes	No

Parts and Attributes

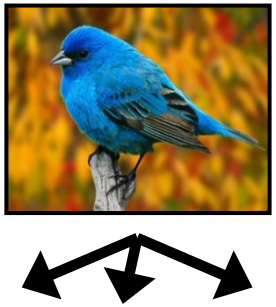
Why use parts and attributes?

They are useful for various reasons:

- Shared by different categories (*e.g.*, birds share the same parts)
- Describing categories
- Can characterize many categories

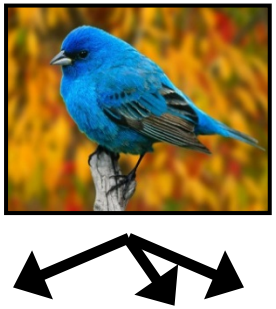
They enable richer types of input from humans.

Hard for Humans



Finch? Bunting?…

Easy for Humans



Yellow Belly? Blue Belly? …

[Branson, Wah *et al*. 2010]

pictureframes.com

STYLE
Traditional

FINISHES
Satin

COLOR
Black

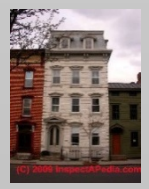
whatBird.com

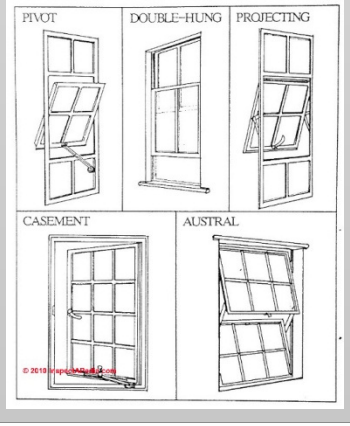
BELLY PATTERN
Striped
Spotted
Solid

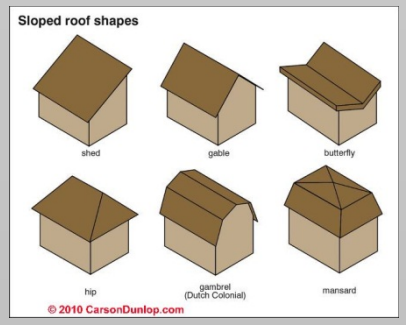
SHAPE
Long-legged
Duck-like
Owl-like
Gull-like

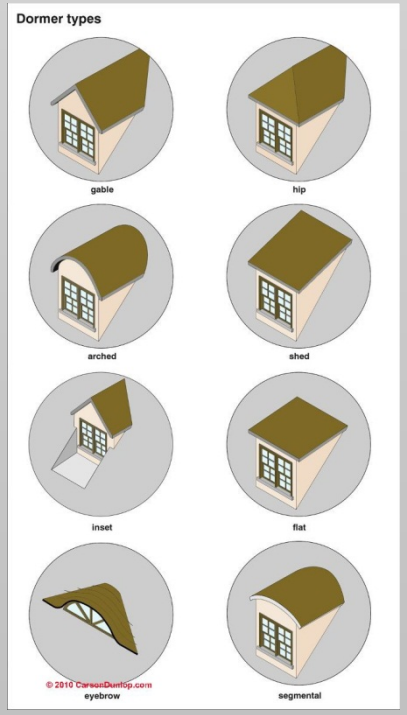
HEAD PATTERN
Spotted
Malar
Crested
Masked

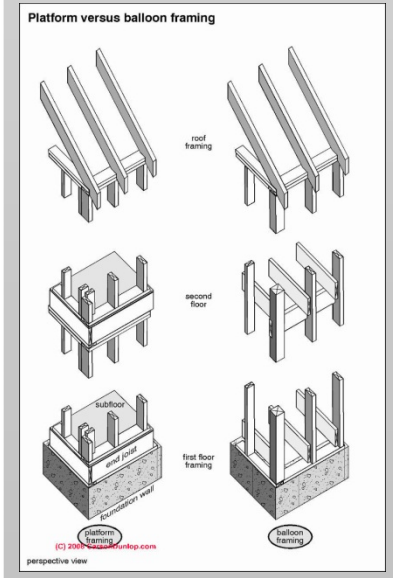
A FIELD GUIDE TO AMERICAN HOUSES





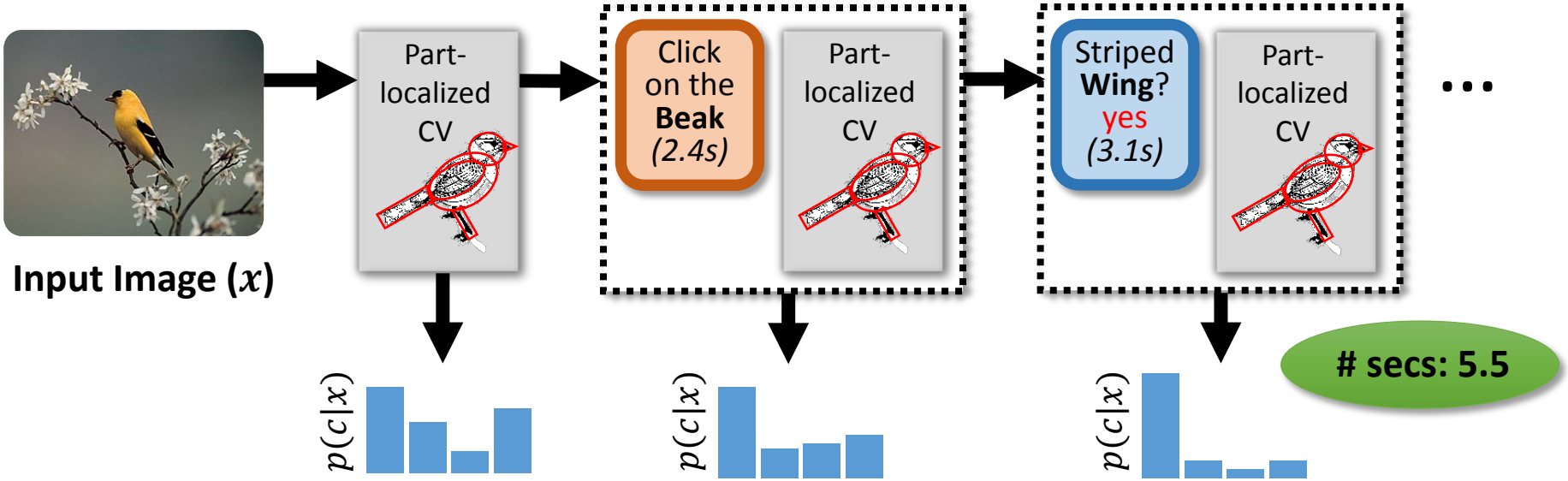


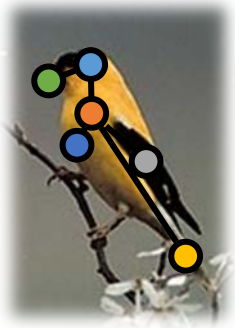




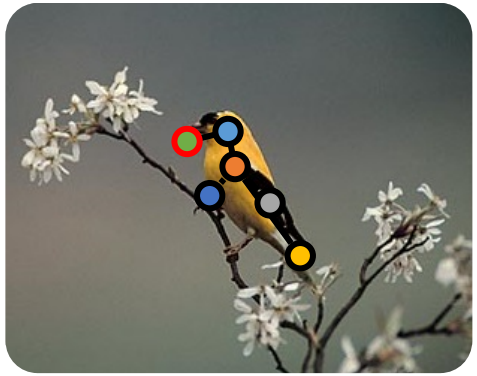
[inspectapedia.com]

Interactive categorization system





Ground Truth Part Locations



[Wah *et al*. 2011]

Attribute question

Part question

Probabilistic model for categorization

Image x

Unique attribute vector $\boldsymbol{a}^c$ for class c

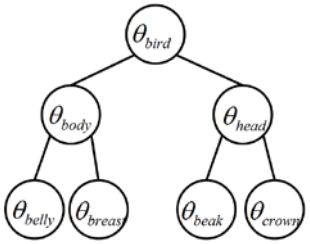
Part locations Θ

User responses U^t at time t

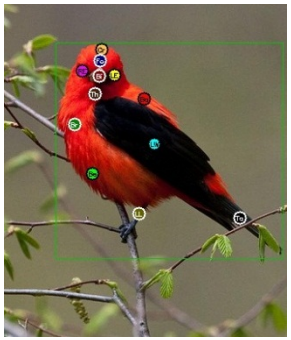
$$p(\boldsymbol{a}^c, U^t, \Theta|x) = \underbrace{p(\boldsymbol{a}^c|\Theta, x)}\underbrace{p(\Theta|x)}\underbrace{p(U^t|\boldsymbol{a}^c, \Theta, x)}$$

Attribute detection

Part detection



How users answer questions



$$p(c|U^t, x) = \frac{p(\boldsymbol{a}^c, U^t|x)}{\sum_c p(\boldsymbol{a}^c, U^t|x)}$$

$$p(\boldsymbol{a}^c, U^t|x) = \int_\Theta p(\boldsymbol{a}^c, U^t, \Theta|x) d\Theta$$

Selecting the next question

To compare part click and binary attribute questions equally:

- We use expected **information gain (IG)**
- Approximated using a random sampling method

We compare **two criteria for question selection**:

Expected IG Criterion

Maximizes expected IG

Minimizes the
total number of questions asked

Time Criterion

Trades off IG relative to expected answer time (sec)

Minimizes the
expected human time spent

Evaluation: details

CUB-200-2011 dataset

- 200 classes of bird species, 11.8K images
- 312 binary attributes, 15 parts

Computer vision

- **Attribute detectors:** linear SVMs on vector-quantized SIFT/color histograms
- **Part detectors:** HOG templates [Felzenszwalb *et al*. 2008]

Evaluation

- Use specified train/test split
- **Sample from stored user responses** at test time
- Assume users **verify the correct class perfectly**
- **Performance metric**: Mean classification accuracy over all test images, after a certain amount of time has elapsed

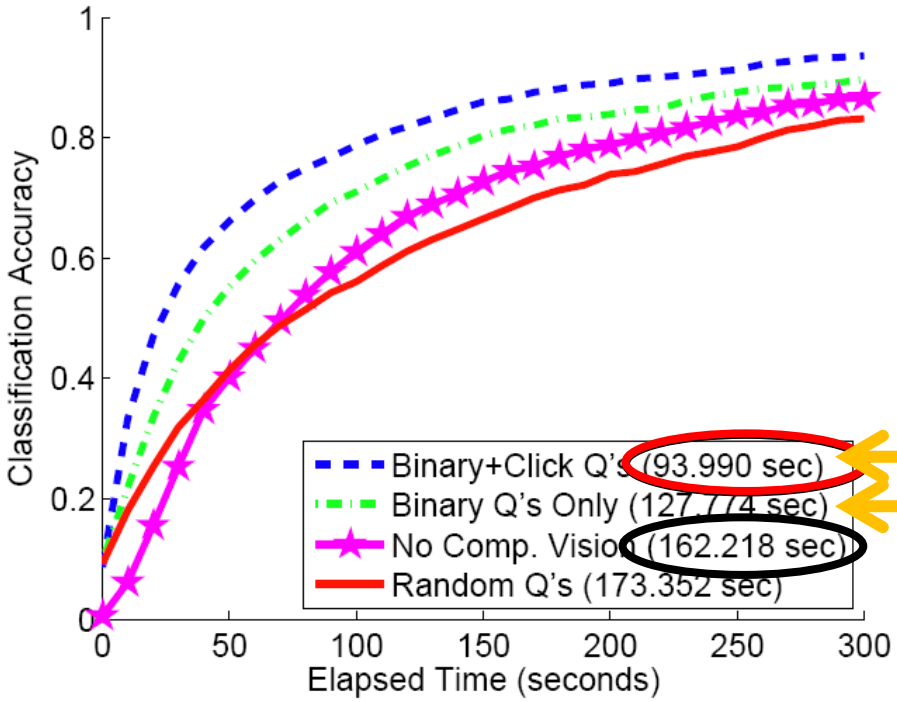
Evaluation: comparison

- **Binary attribute and part click** questions [Wah *et al*. 2011]
- **Binary attribute** questions only [Branson, Wah *et al*. 2010]
- **No computer vision**
- **Selecting questions at random**

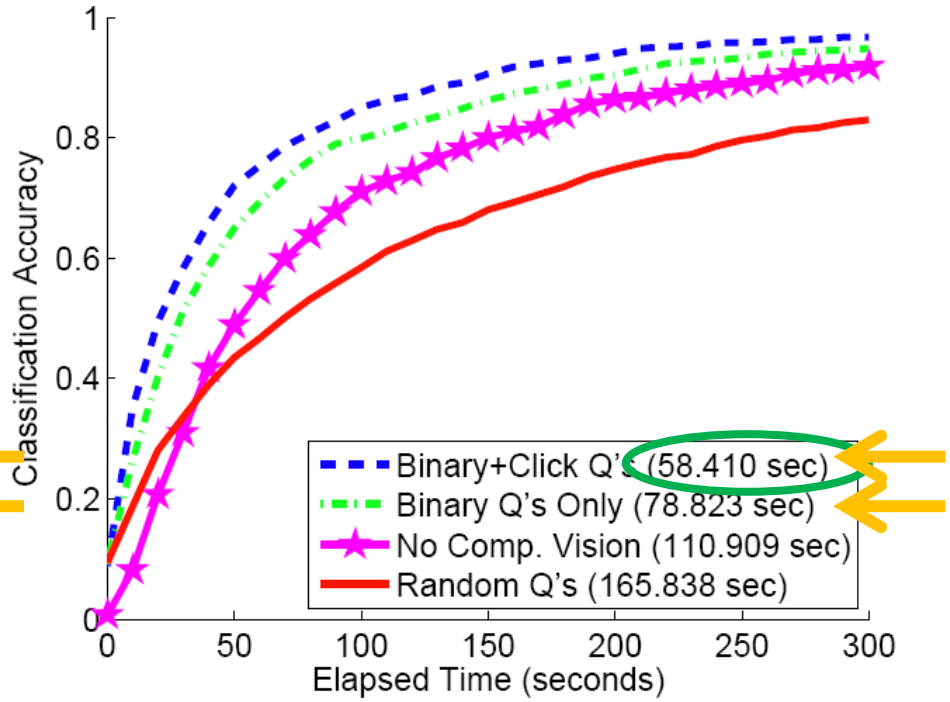
Using computer vision reduces the burden on the user.

By localizing parts, we improve performance.

Expected IG Criterion



Time Criterion

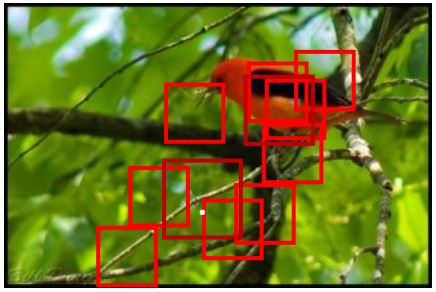


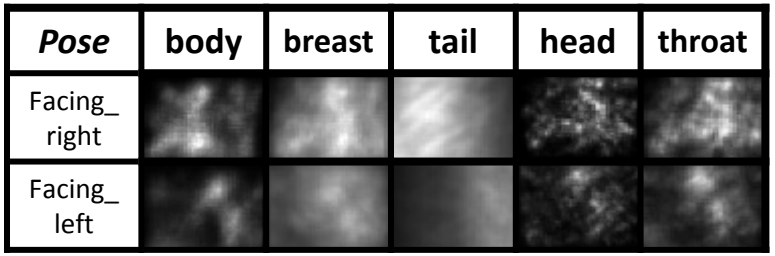
Choosing questions by time reduces human effort.

Example results

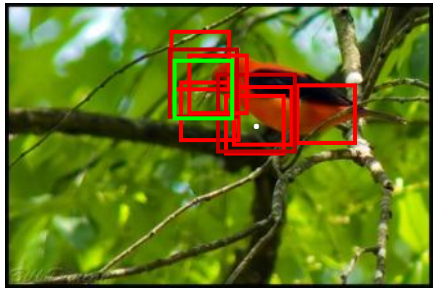
As the user answers questions, we can localize the bird and estimate its pose more accurately.

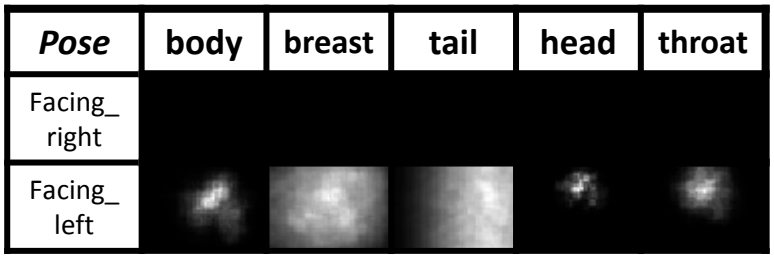
Initial predicted part locations





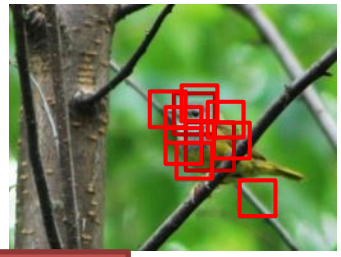
Q1: Click on the **beak** *(3.200 s)*





By using a combination of attribute and part click questions, we can identify the correct class.

Initial predicted part locations



Orange-Crowned Warbler? **no**

Q1: Is the throat yellow? **no** *(4.547 s)*

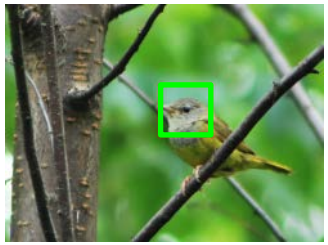
Great Crested Flycatcher? **no**

Q2: Click on the **tail** *(2.043 s)*

Yellow Bellied Flycatcher? **no**

Q3: Click on the **head** *(2.452 s)*

Mourning Warbler? **yes**



Parts and attributes: summary

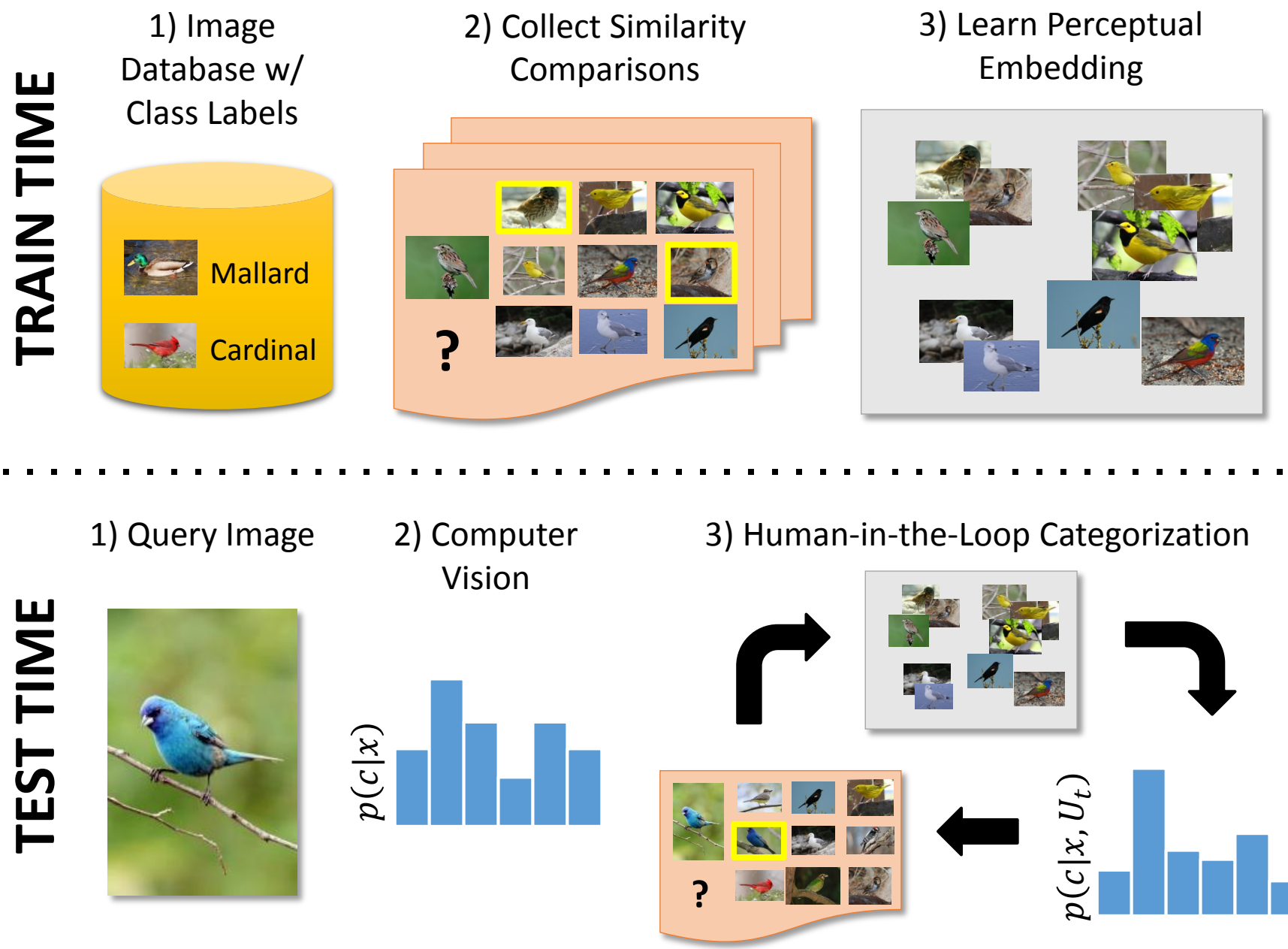
- We use **class attribute and part models** to perform interactive categorization
- We evaluate different forms of user input and **intelligently choose the next question** to ask
- **Scales efficiently** to large numbers of categories

Similarity Learning

Parts and attributes aren't perfect

- Drawbacks:
  - Scalability
  - Costly
  - Relies on experts
  - Experts are scarce

By using similarity instead, we can reduce our dependence on expert-derived vocabularies.



[Wah *et al*. 2014]

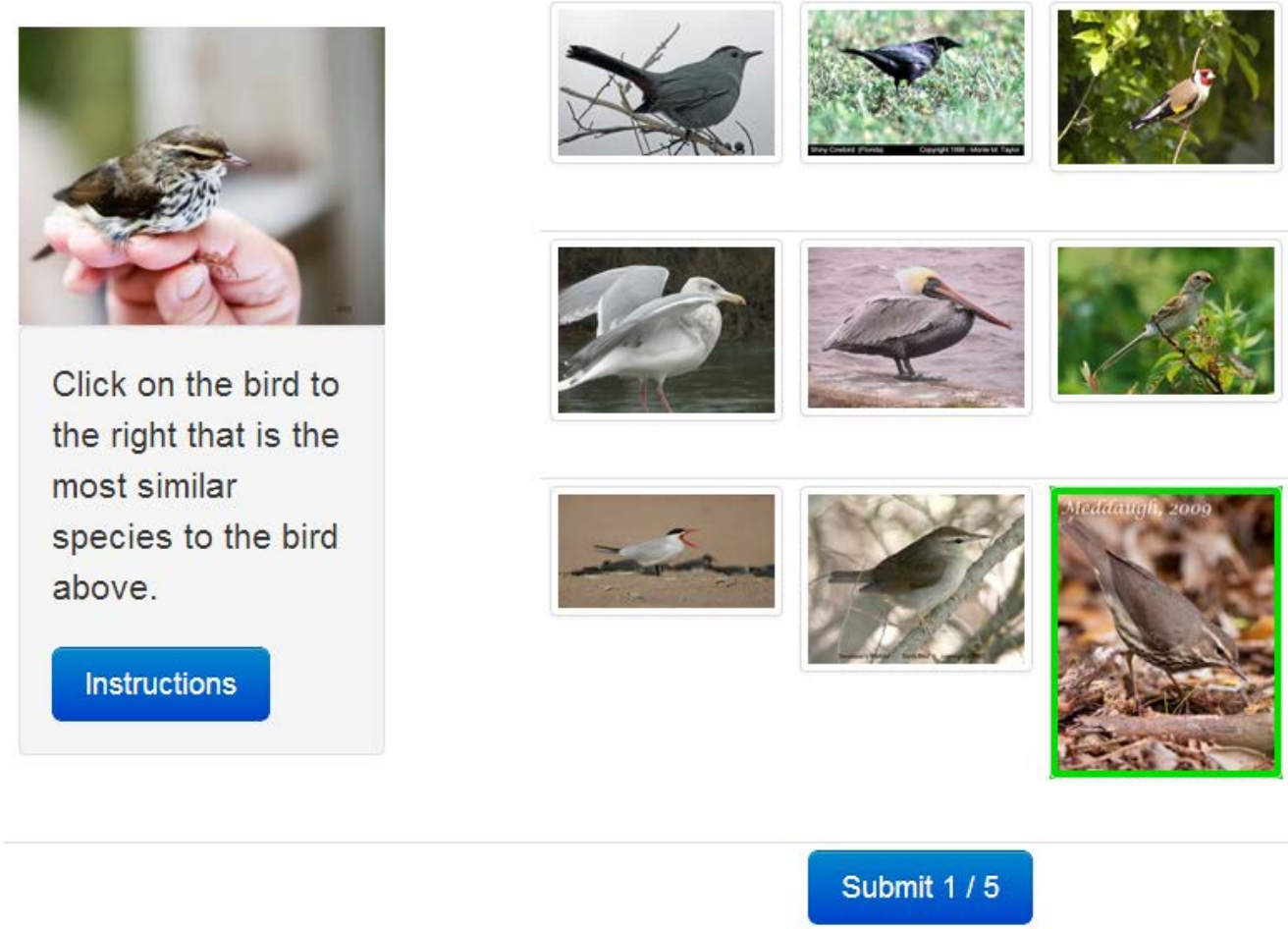
Relative similarity comparisons

Is this more similar to…

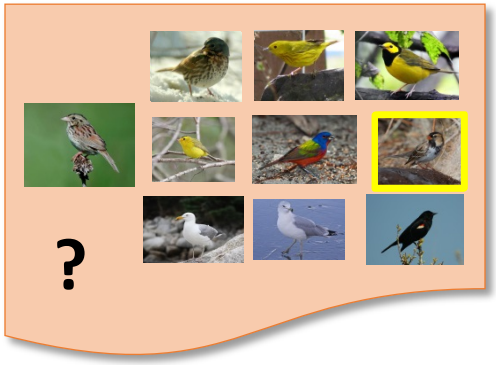
This one?

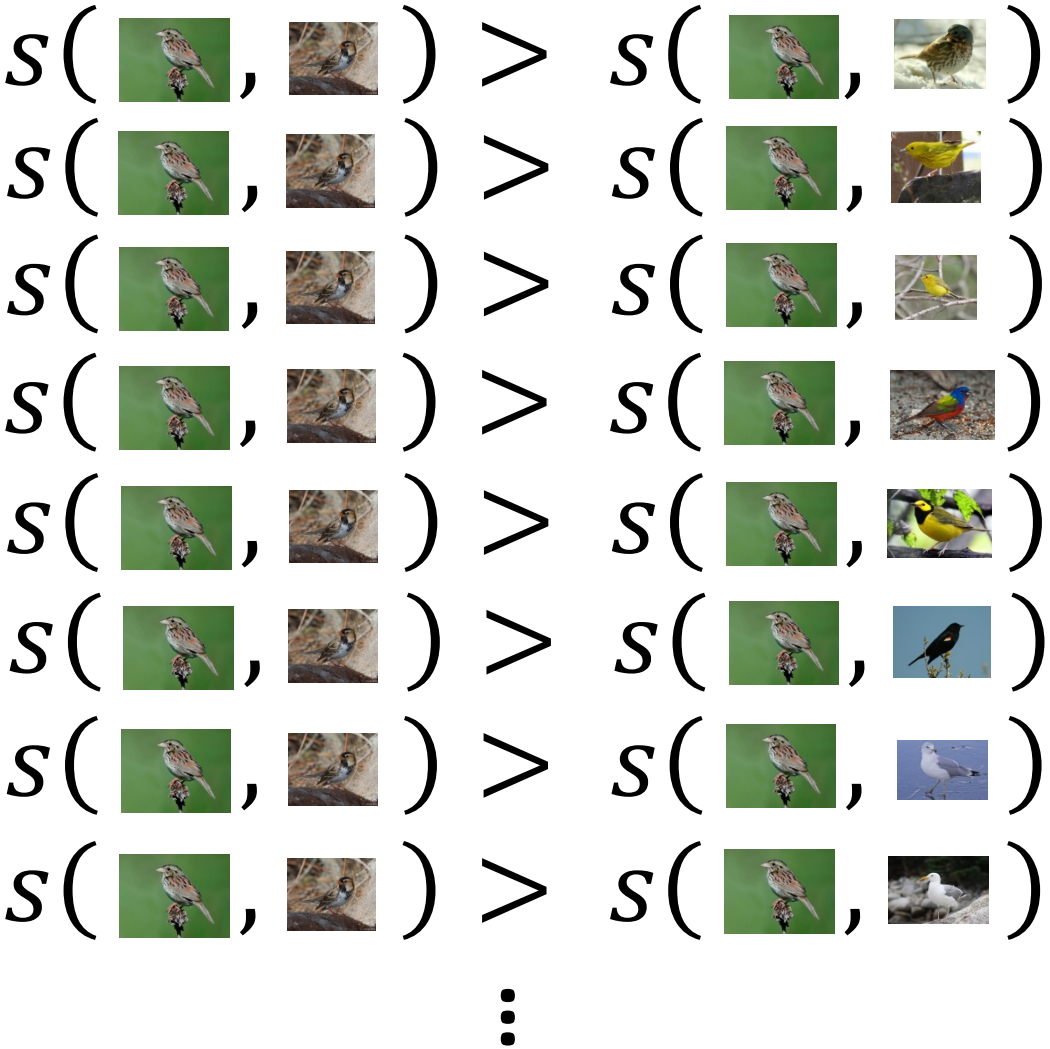
Or this one?

Relative similarity comparisons



Generating triplet comparisons





$s(i,j)$: perceptual similarity between images x_i and x_j

Learning a similarity metric

$$\mathcal{T} = \left\{(i,j,l) \middle| x_i \text{ is more similar to } x_j \text{ than } x_l\right\}$$

With $\mathcal{T}$, we learn an **embedding** $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N]$

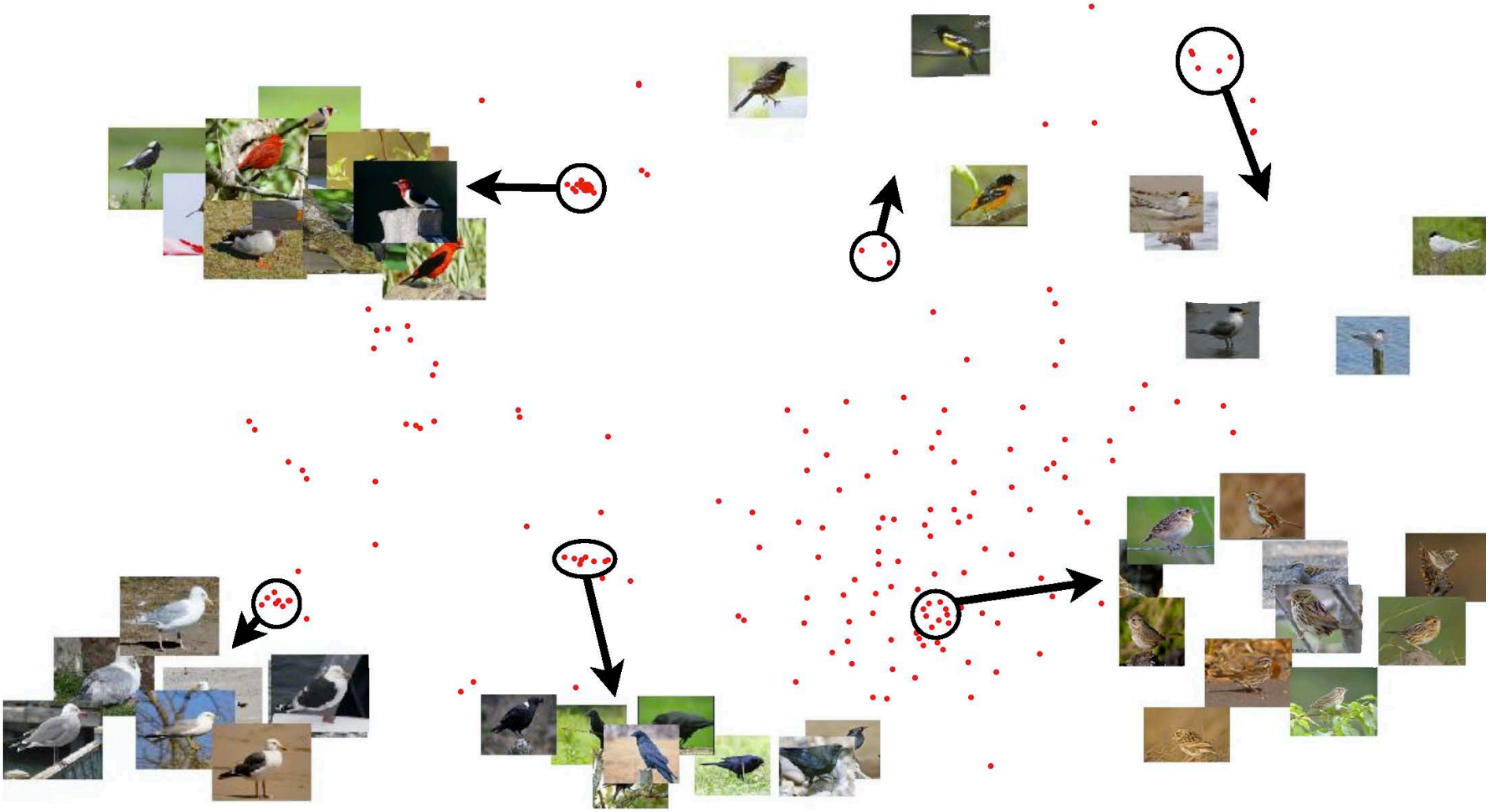
- of N training images
- using **stochastic triplet embedding**

From $\mathbf{Z}$ we generate a **similarity matrix** $S \in N \times N$, where

$$S_{ij} = \exp\left(-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{2\sigma^2}\right)$$

[van der Maaten & Weinberger 2012]

Perceptual similarity embedding over 200 categories



Perceptual similarity embedding over 200 categories

Interactive categorization

x	Query image
c	Class
t	Time step
U_t	User responses at t
$\mathbf{z}$	True location of x in perceptual space

Computing per-class probabilities:

$$p(c, |x, U_t) \propto p(c, U_t|x) = \int_{\mathbf{z}} p(c, \mathbf{z}, U_t|x) d\mathbf{z}$$

User responses at timestep t

Models how users respond to similarity questions

Computer vision estimate

Location of query image x in perceptual space

$$w^t = p(c, \mathbf{z}, U_t|x) = \overbrace{p(U_t|\, c, \mathbf{z}, x)}\, \overbrace{p(c, \mathbf{z}|x)}$$

Incorporating a user response (D is the grid of images for each question):

$$p(u|\mathbf{z}) = \frac{\phi(s(\mathbf{z}, \mathbf{z}_i))}{\sum_{j \in D} \phi\big(s(\mathbf{z}, \mathbf{z}_j)\big)}$$

Incorporating computer vision, e.g., from a multiclass classifier:

$$p(c, \mathbf{z}|x) \propto p(c|x)$$

Efficient

Interactive categorization

x	Query image
c	Class
t	Time step
U_t	User responses at t
$\mathbf{z}$	True location of x in perceptual space

Computing per-class probabilities:

$$p(c, |x, U_t) \propto p(c, U_t|x) \approx \frac{\sum_{k, c_k = c} w_k^t}{\sum_k w_k^t}$$

Weight w_k of training example k represents how likely $\mathbf{z}_k$ is the true location $\mathbf{z}$:

$$w_k^t = p(c_k, \mathbf{z}_k, U_t|x) = p(U_t|\, c_k, \mathbf{z}_k, x)\, p(c_k, \mathbf{z}_k|x)$$

Such that:

$$w_k^{t+1} = p(u_{t+1}|\mathbf{z}_k) w_k^t$$

$$= \frac{\phi(S_{ik})}{\sum_{j \in D} \phi(S_{jk})} w_k^t$$

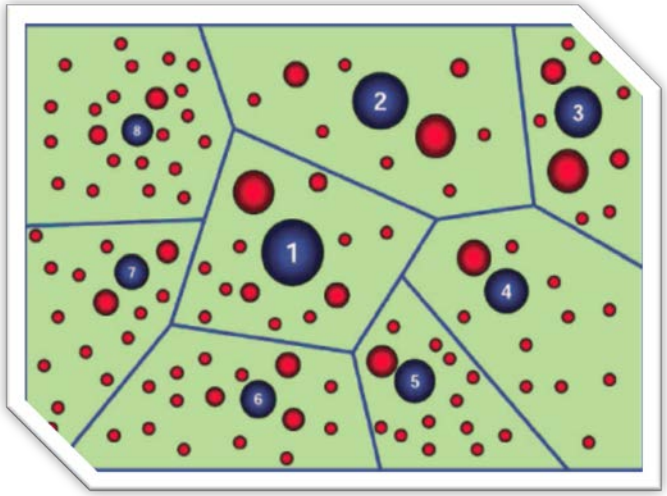
Efficient update rule:

1. Initialize weights $w_k^0 = p(c_k, \mathbf{z}_k|x)$
2. Update weights w_k^{t+1} when user answers a similarity question
3. Update per-class probabilities

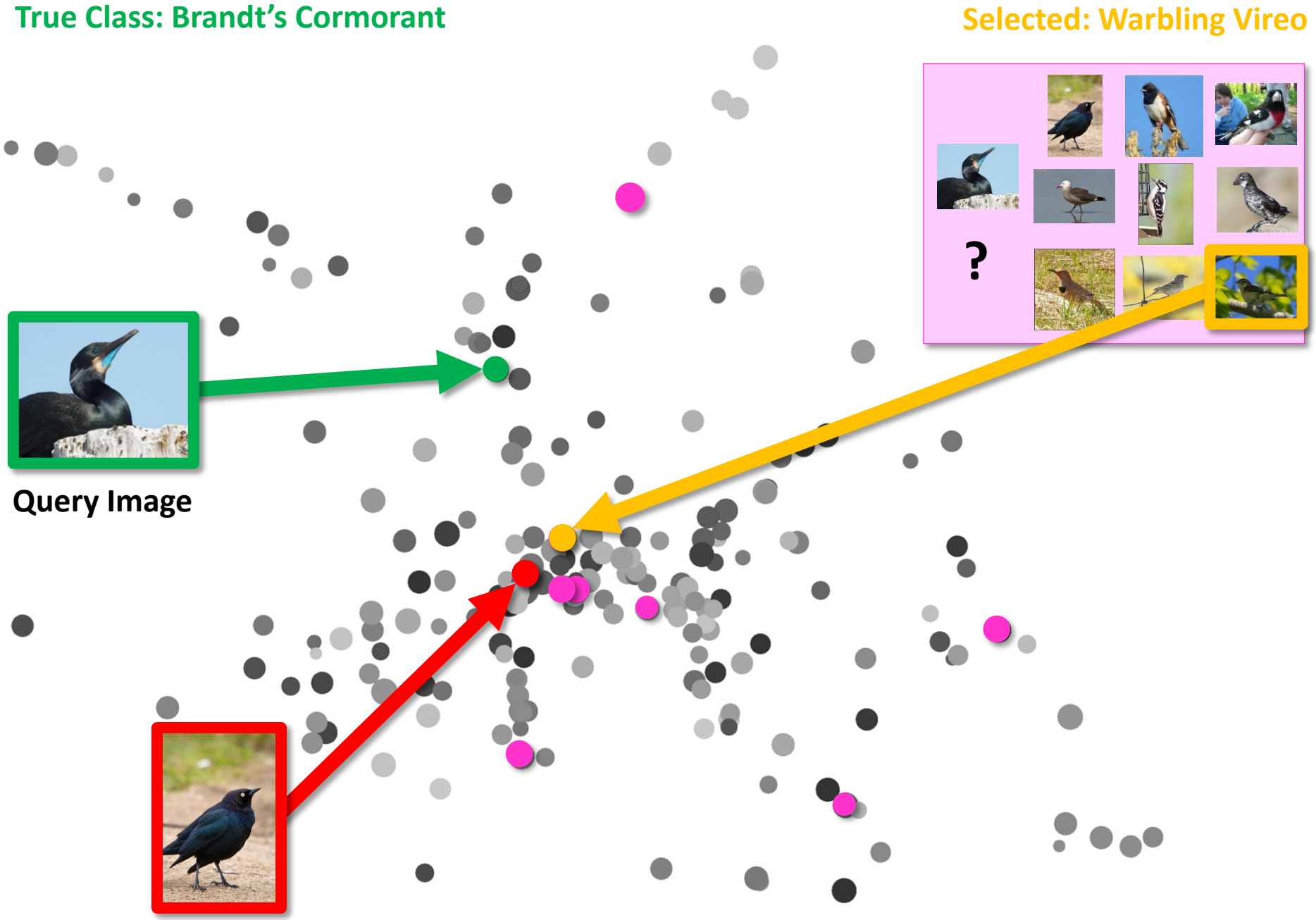
Selecting images for the display

Approximate solution

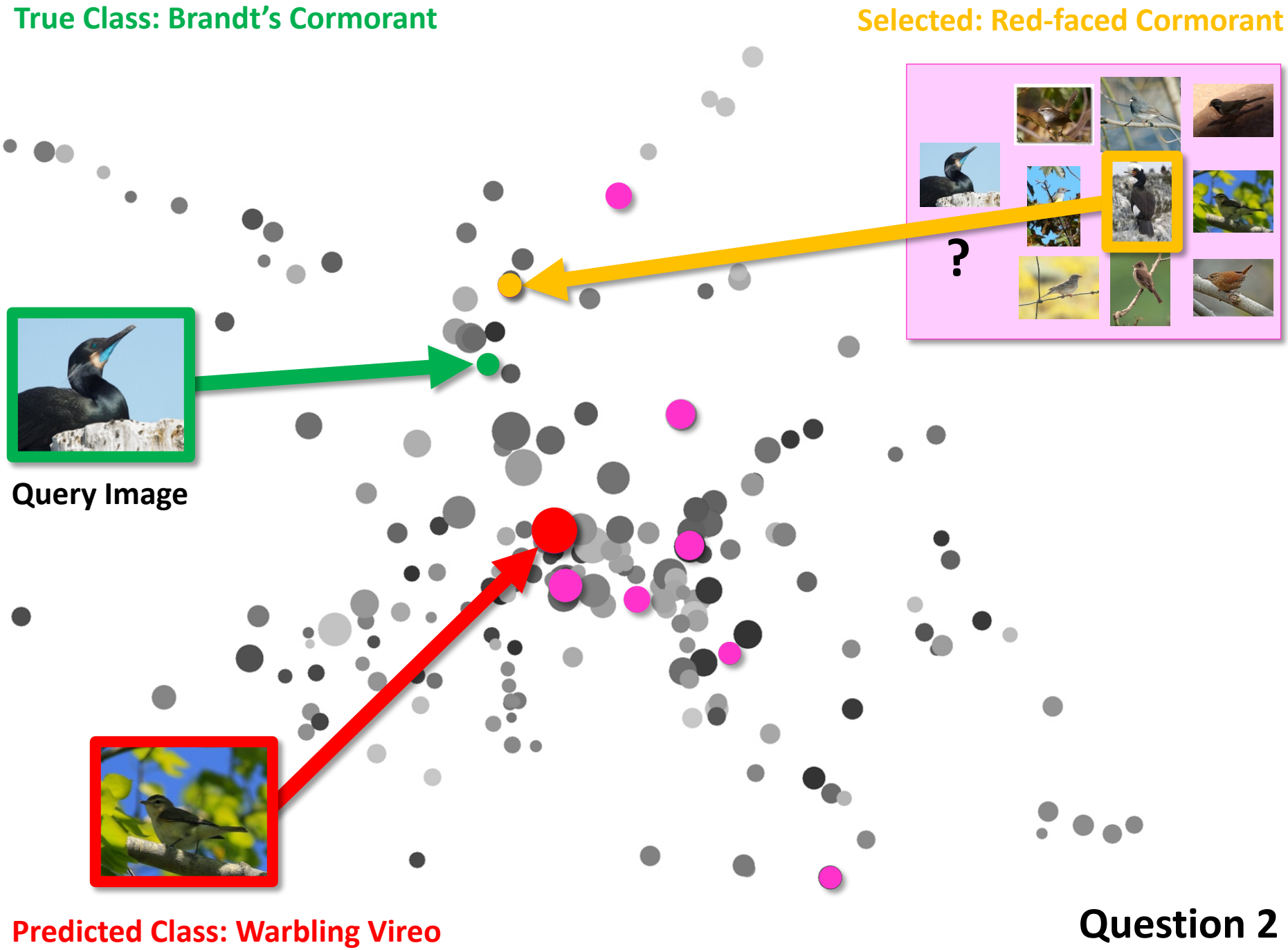
- Group images into equal-weight clusters
- From each cluster, select the image with largest w_k^t
- Maximizes expected IG for $p(c, \mathbf{z}_k, U_t | x)$

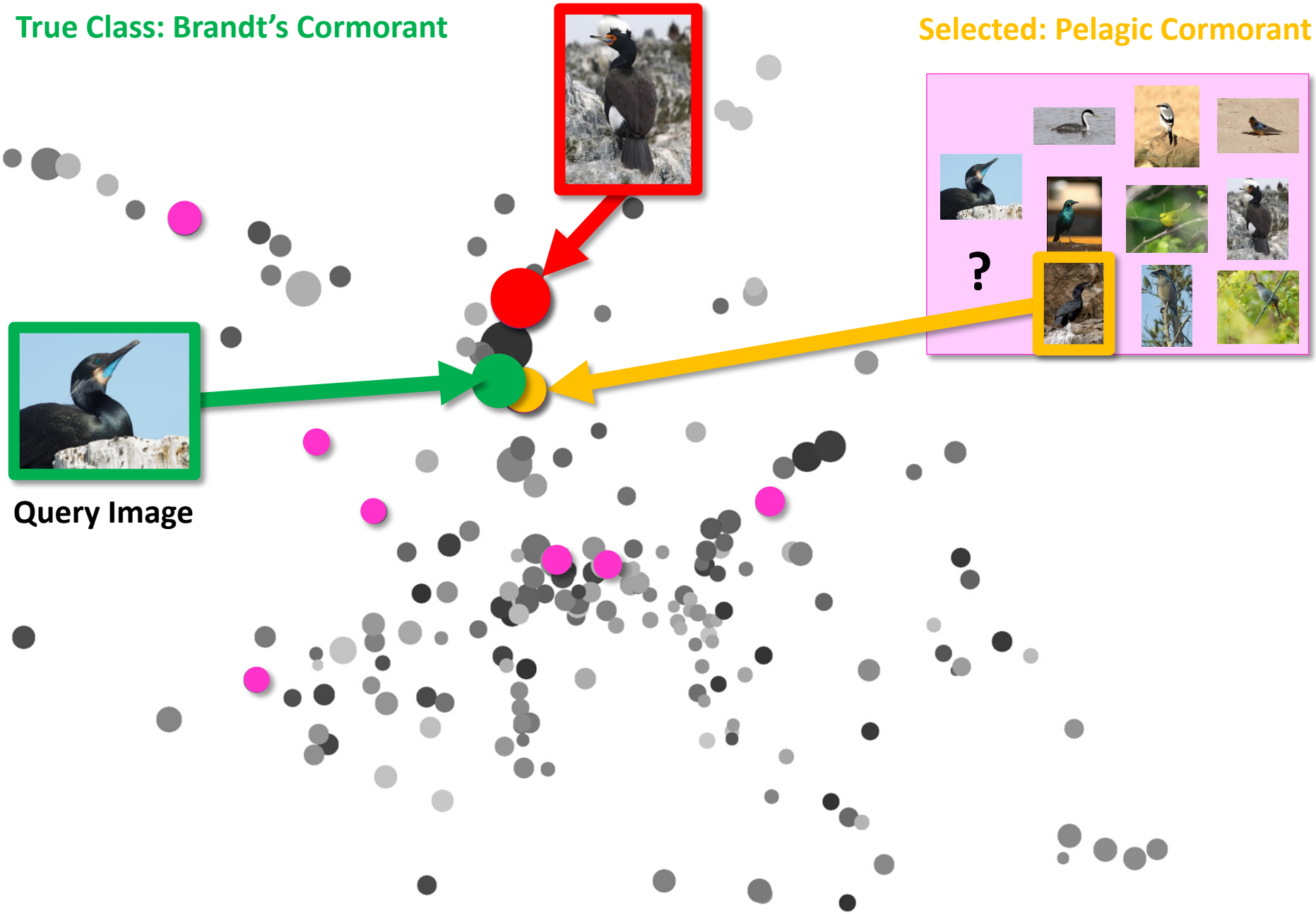


[Fang & Geman 2005, Ferecatu & Geman 2009]

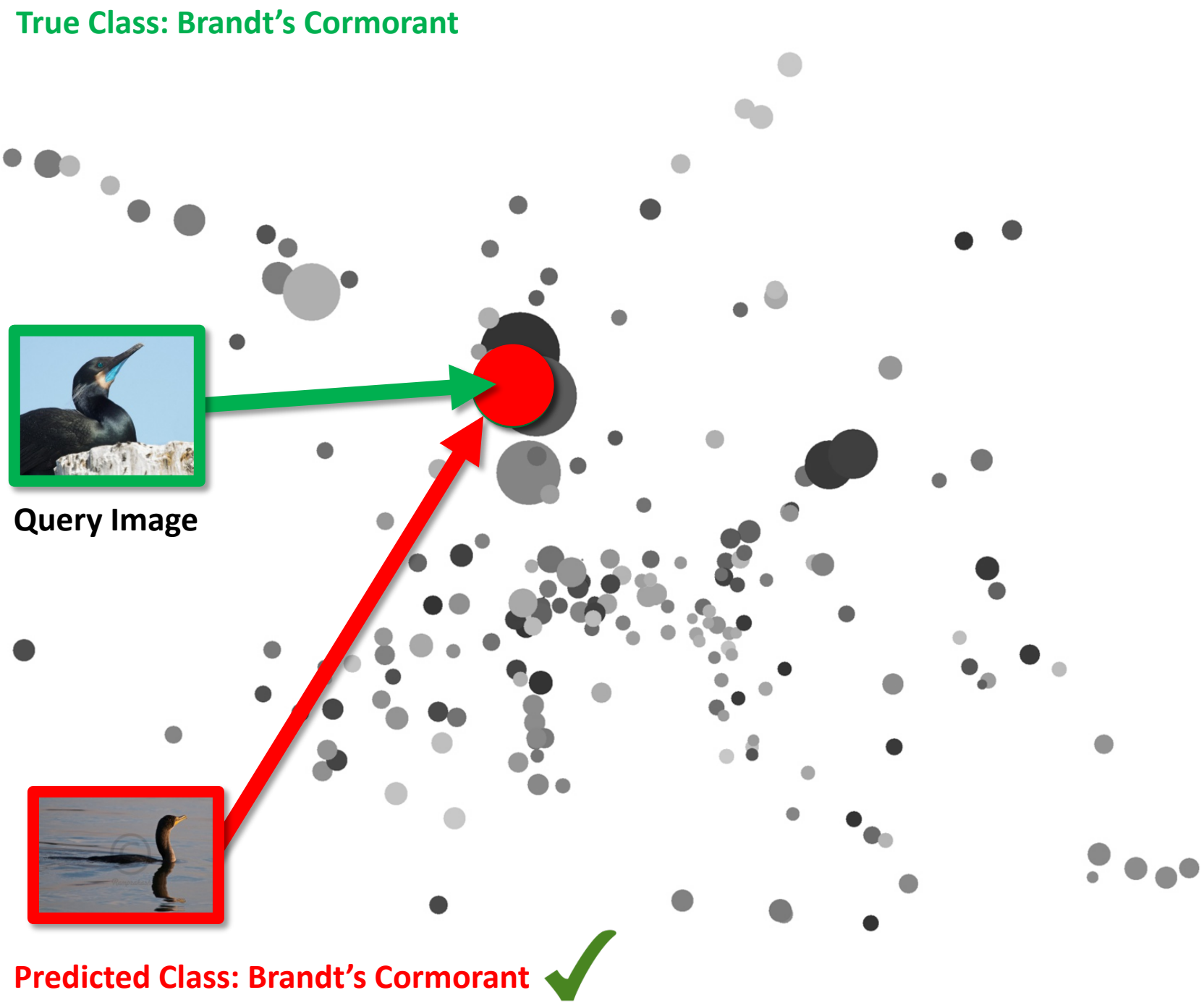

True Class: Brandt's Cormorant
Selected: Warbling Vireo
?
Query Image
Predicted Class: Brewer's Blackbird

Question 1


True Class: Brandt's Cormorant
Selected: Red-faced Cormorant
?
Query Image
Predicted Class: Warbling Vireo
Question 2


True Class: Brandt's Cormorant
Selected: Pelagic Cormorant
?
Query Image
Predicted Class: Red-faced Cormorant

Question 3


True Class: Brandt's Cormorant
Query Image
Predicted Class: Brandt's Cormorant

Wait a minute…

- Similarity is not always easy to judge
  - **Multiple dimensions** (color, shape, pattern, *etc*.)
  - **Nuisance factors** (pose, background, illumination, *etc*.)

It is easier for a human to judge localized similarity.

Which of the two images is more similar to this image?

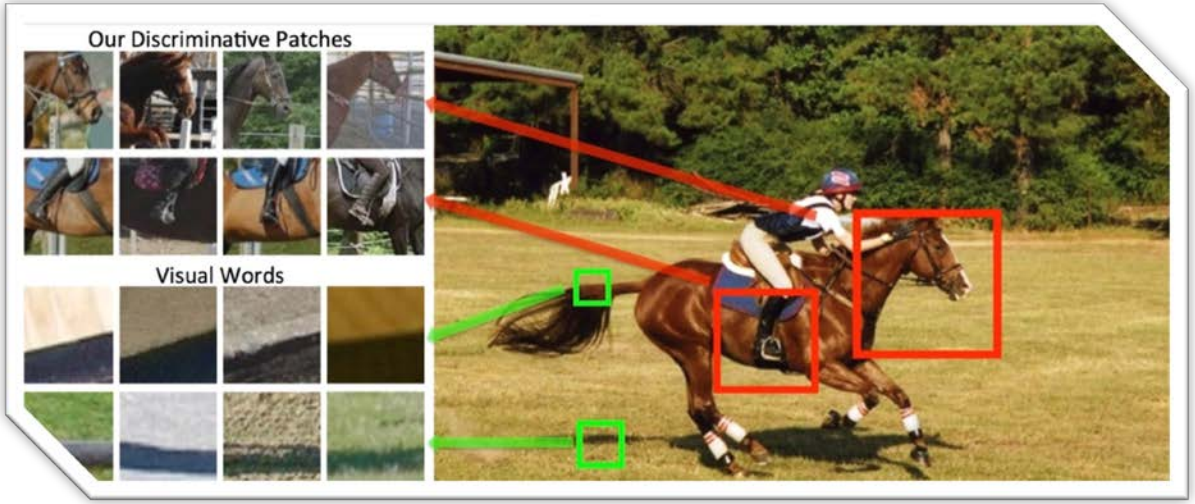
Nonlocalized similarity comparison

Localized similarity comparison

Learning localized metrics

- Discover localizable, discriminative regions
- Collect similarity comparisons for localized regions
- Learn multiple independent metrics

"Unsupervised discovery of mid-level discriminative patches"

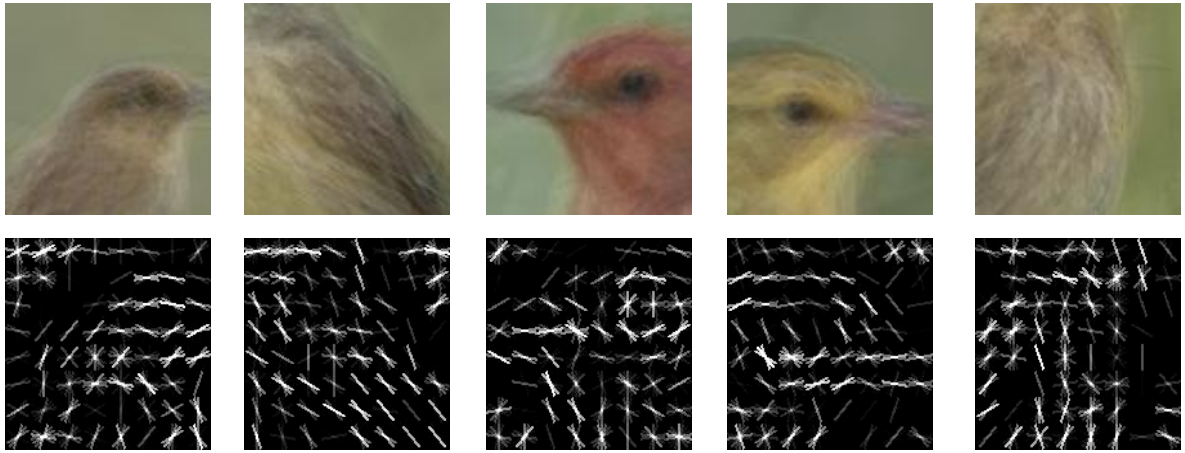


[Singh *et al*. 2012]

[Wah *et al*. 2014, under review]

Learning localized metrics

- Discover localizable, discriminative regions
- Collect similarity comparisons for localized regions
- Learn multiple independent metrics

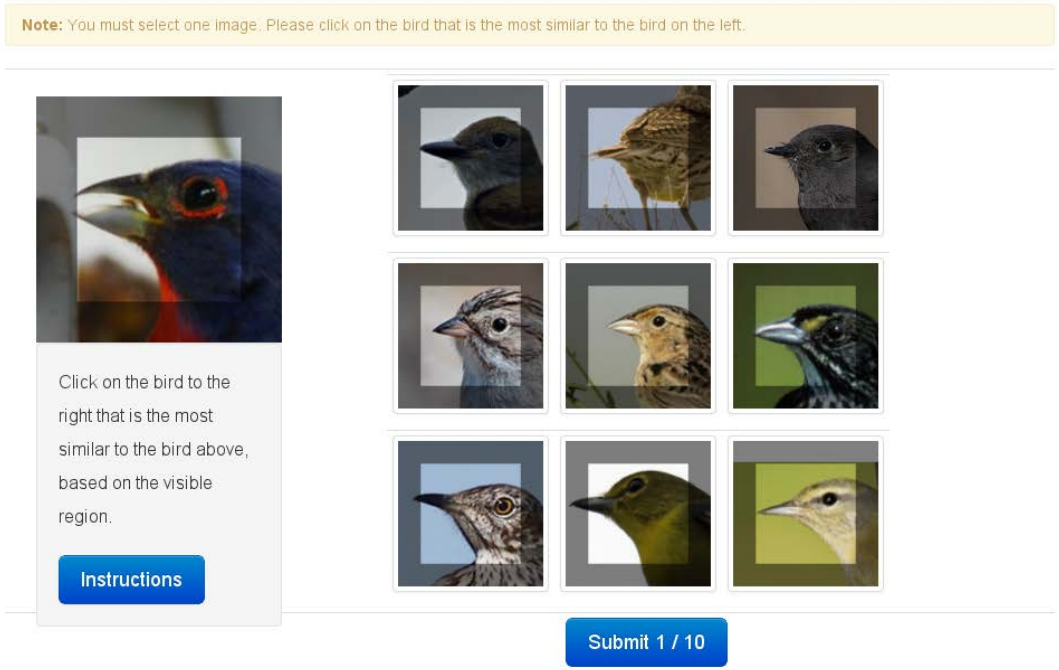


1 13 21 23 39

[Wah *et al*. 2014, under review]

Learning localized metrics

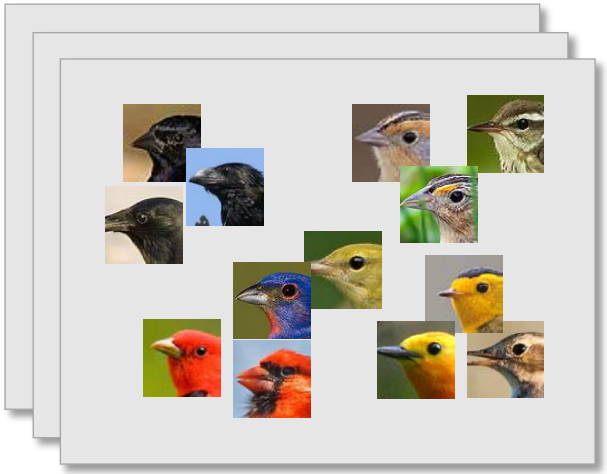
- Discover localizable, discriminative regions
- Collect similarity comparisons for localized regions
- Learn multiple independent metrics



[Wah *et al*. 2014, under review]

Learning localized metrics

- Discover localizable, discriminative regions
- Collect similarity comparisons for localized regions
- Learn multiple independent metrics



Similarity matrices $S^1 \ldots S^R$
Region $r \in \mathcal{R}$ where $\mathrm{R} = |\mathcal{R}|$

[Wah *et al*. 2014, under review]

Localized interactive categorization

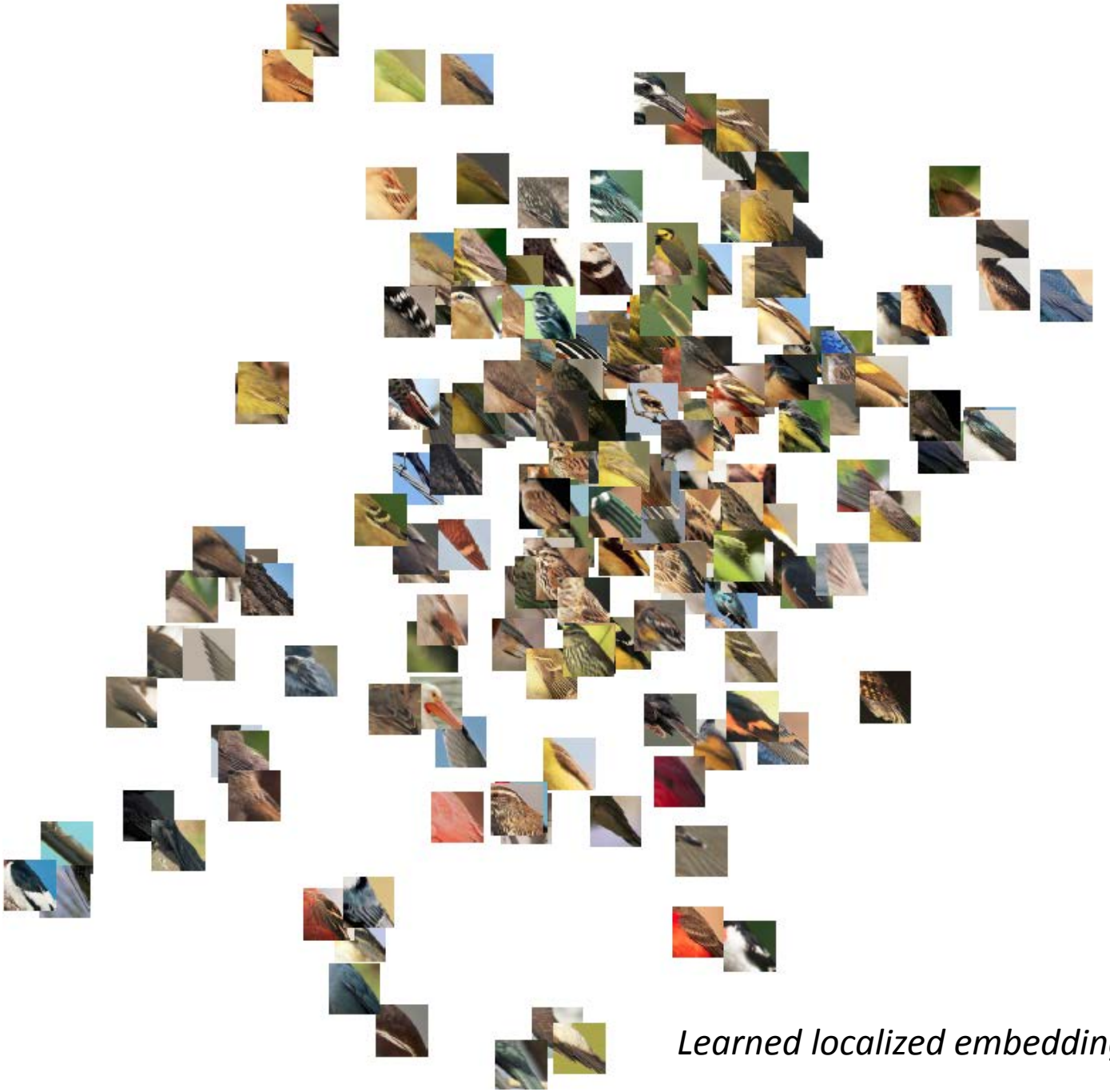
Selecting images for the display

- **Problem**: region r is not present/visible in all images
- **Solution**: show images most likely to have r present
  - For each r, use **approximate clustering solution** as before

Selecting the next question/region

- **Solution**: select r that produces the most balanced clustering, *i.e.*, maximizes expected information gain
  - Update weights w_k^{t+1} according to similarity matrix S_r of selected region r

[Fang & Geman 2005, Ferecatu & Geman 2009]



Learned localized embedding for

Evaluation: details

Region discovery

- 106 discovered; 5 selected for experiments

Computer vision

- Multiclass classifiers with **SIFT/color histograms** with spatial pyramids (*19.4% acc*)
- Linear SVMs on SIFT descriptors with **Fisher vector encodings** (*34.8% acc*)

Evaluation

- Use specified train/test split
- **Probabilistic user model** to simulate responses at test time
- Assume users **verify the correct class perfectly**
- **Performance metric**: Mean classification accuracy over all test images, after a certain number of questions have been asked

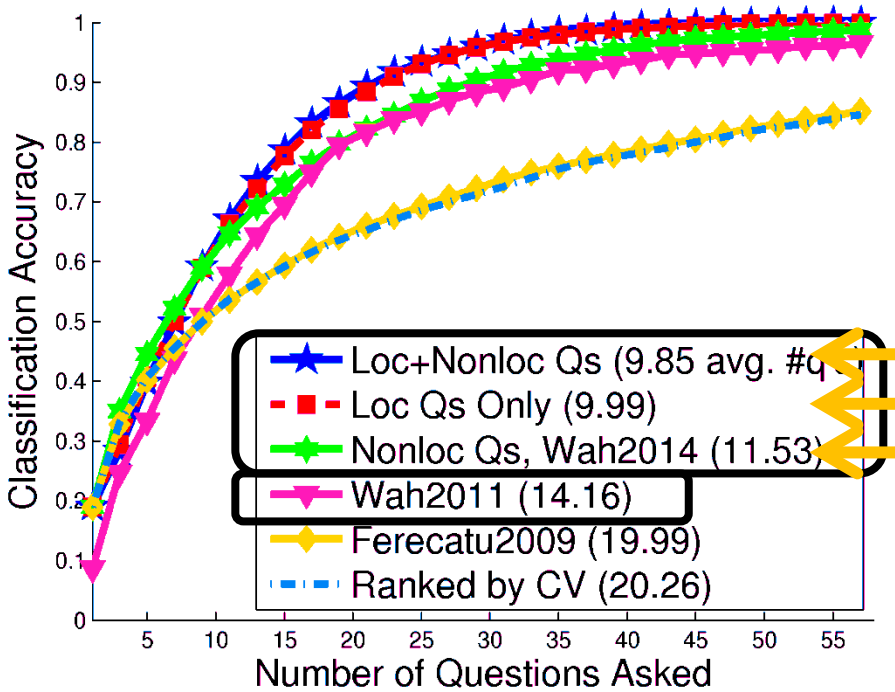
Evaluation: comparison

- **Perceptual similarity**
  - **Localized** metrics [Wah *et al*. 2014, under review]
  - **Nonlocalized** metrics [Wah *et al*. 2014]
  - Both **nonlocalized and localized** metrics
- **Relevance feedback**-based framework (*modified*) [Ferecatu & Geman 2009]
- **Binary attribute** questions only [Branson, Wah *et al*. 2010]
- **Binary attribute and part click** questions [Wah *et al*. 2011]
- **Ranked by computer vision** baseline

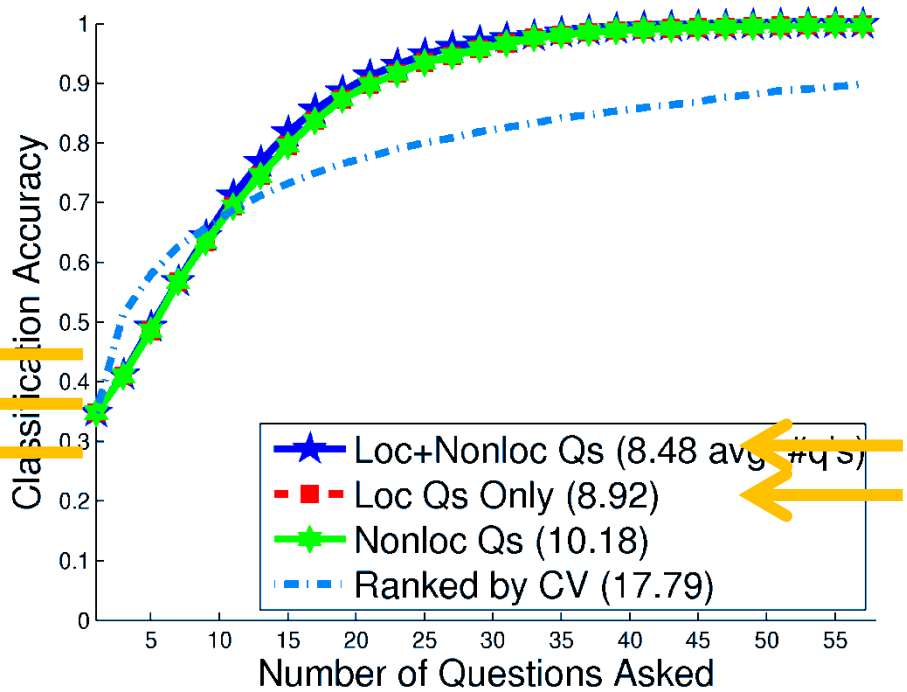
Similarity comparisons outperform part and attribute-based methods.

Using both localized and nonlocalized metrics together helps.

CV: SIFT/color histograms

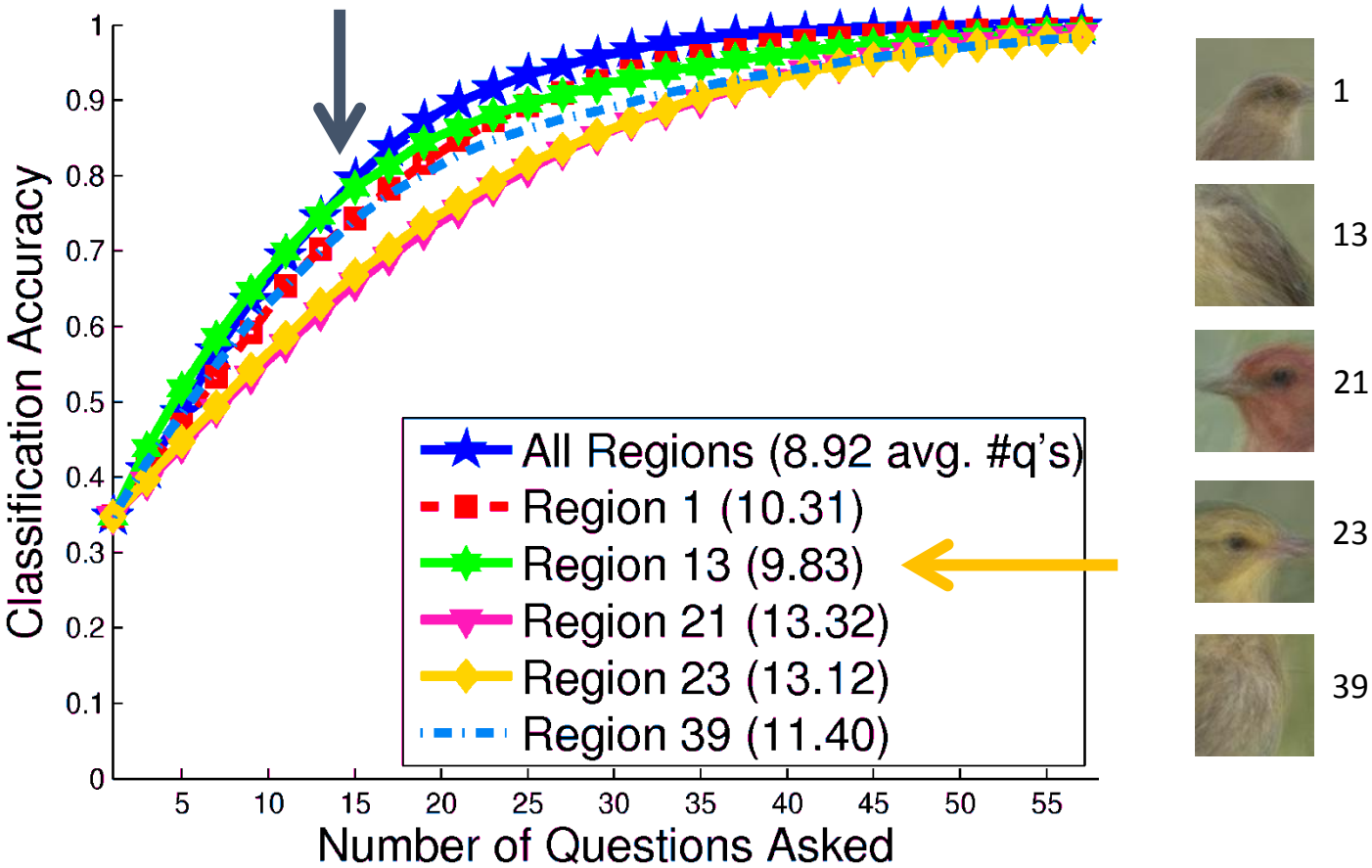


CV: Fisher Vector encoding



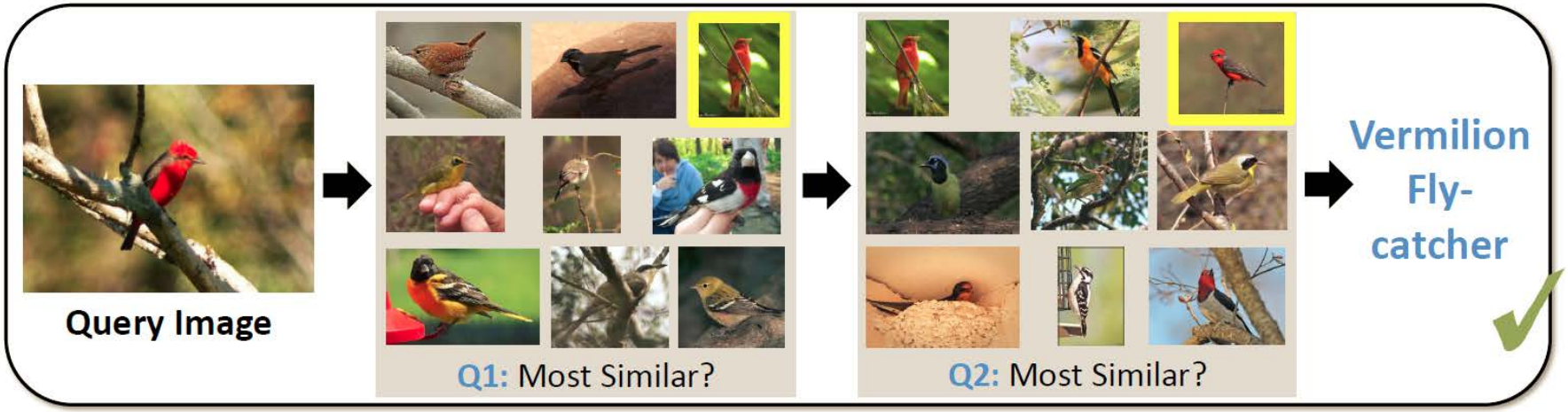
Our results are consistent when we use improved computer vision algorithms.

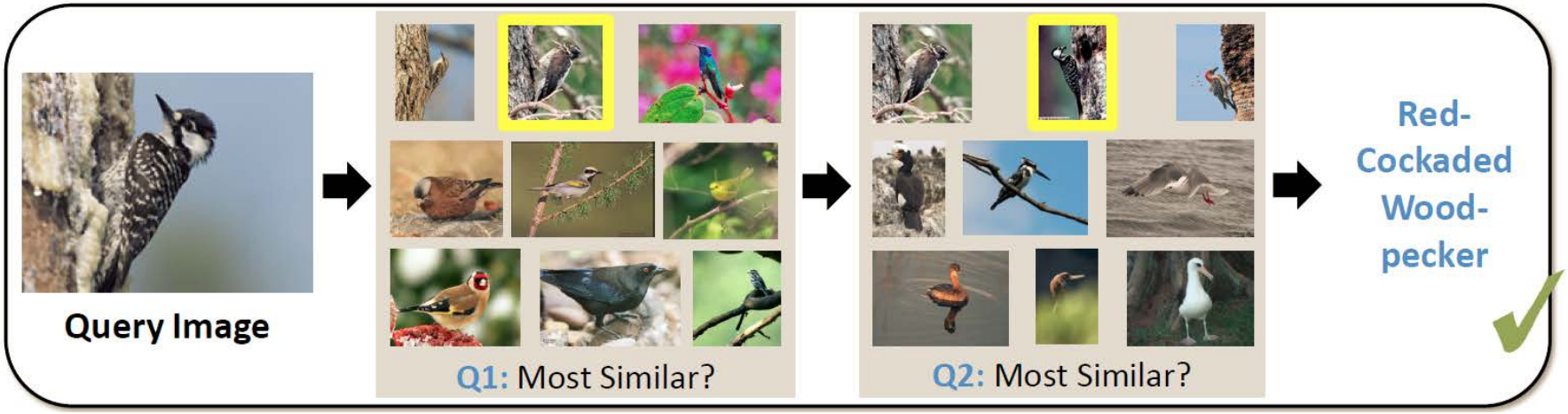
CV: Fisher Vector encoding



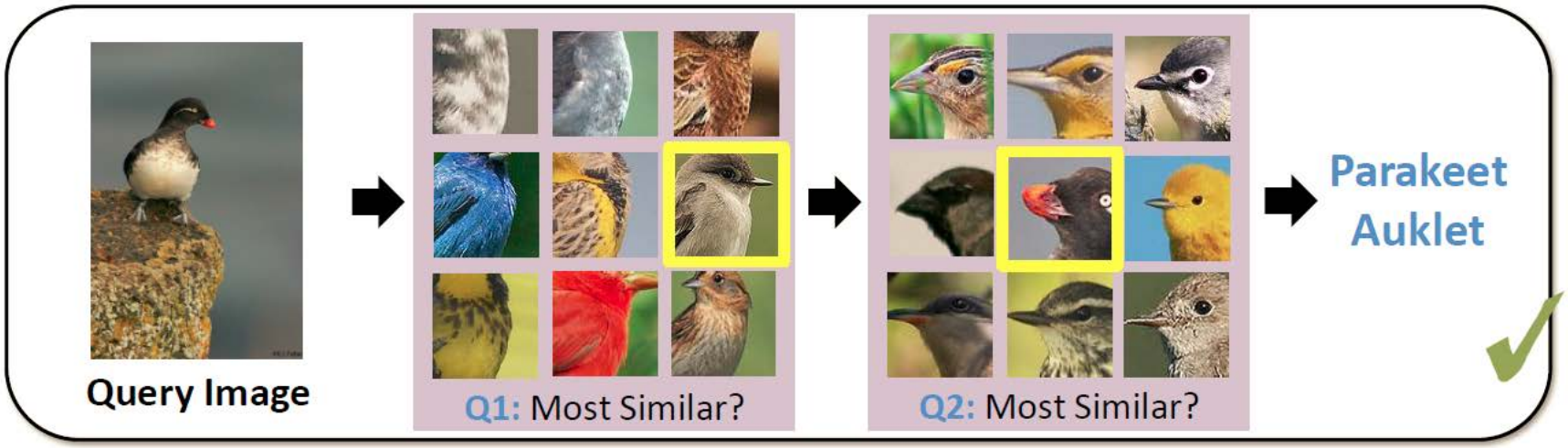
Some localized regions are more useful than others.

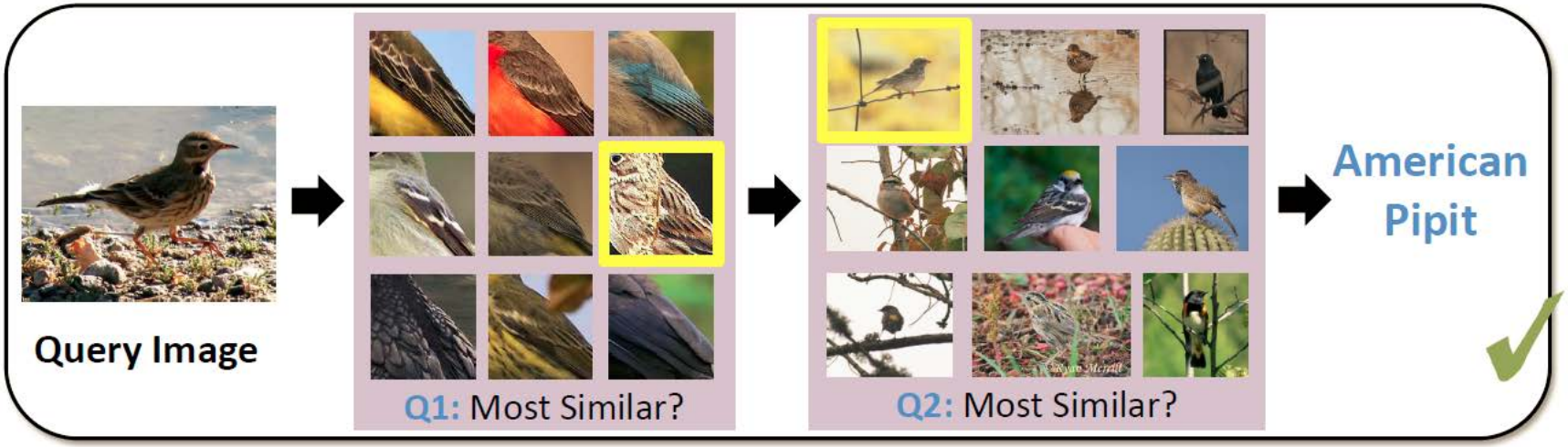
Example results: nonlocalized





Example results: localized





Similarity learning: summary

- We use **perceptual measurements of similarity** to improve performance over part and attribute-based methods

- **Localized similarity**
  - **Easier** for the user
  - **More informative** for the system

- **Easy integration** with different CV algorithms

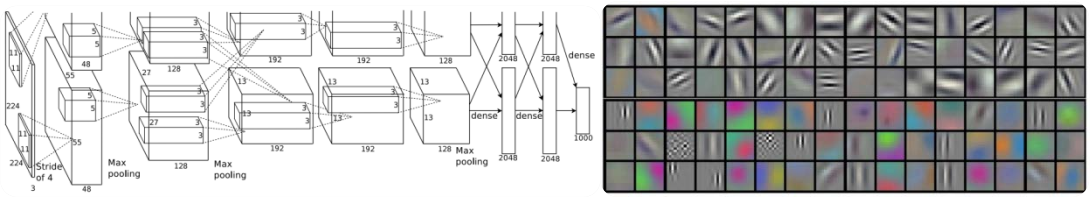
- **Cost-effective to deploy** for new basic-level categories

Future

Future directions

- Applying these paradigms to **new basic-level categories**
- Investigating **best practices for interfacing with users**: creating annotation tasks and querying users for input
- **Automating deployment** of interactive categorization pipelines

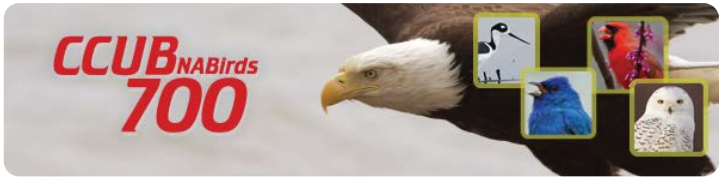
Convolutional neural networks



[Krizhevsky *et al*. 2012]

Birds and beyond

NABirds



Food

[Wilber *et al*. 2014, under review]

Handbags

ebay

and more…

Conclusion

- **Interactive categorization** enables us to leverage the respective strengths of humans and machines
- We can use **part and attribute vocabularies** to efficiently query users and categorize fine-grained objects
- **Relative similarity comparisons** are a powerful way to make the most of human non-experts

Acknowledgements

Family • Christine Wah • Benjamin Wah • Elaine Wah

Advisor • Serge Belongie

Collaborators • Subhransu Maji • Pietro Perona • Steve Branson • Grant Van Horn

Committee • David Kriegman • Gert Lanckriet • Lawrence Saul • Nuno Vasconcelos

Visipedia • Ron Appel • Peter Welinder • Florian Schroff • Ryan Farrell

Vision Lab • Boris Babenko • Kai Wang • Sam Kwak • Tsung-Yi Lin • Oscar Beijbom • Vincent Rabaud • Piotr Dollar • Carolina Galleguillos • Ana Cristina Murillo • Hani Altwaijry • Mohammad Moghimi • Zak Murez • Phuc Nguyen • Arturo Flores • Eric Christiansen • Jan Jakeš • Tomas Matera • Michael Wilber

UCSD • Marti Motoyama • Ryan Braud • Neha Chachra • Chris Kanich • Ryan Kanoknukulchai • Alan Leung • Greg Long • Dan Moeller • Nima Nikzad • Laura Pina • Alex Tsiatas • Nakul Verma • Ming Wang • Tess Winlock

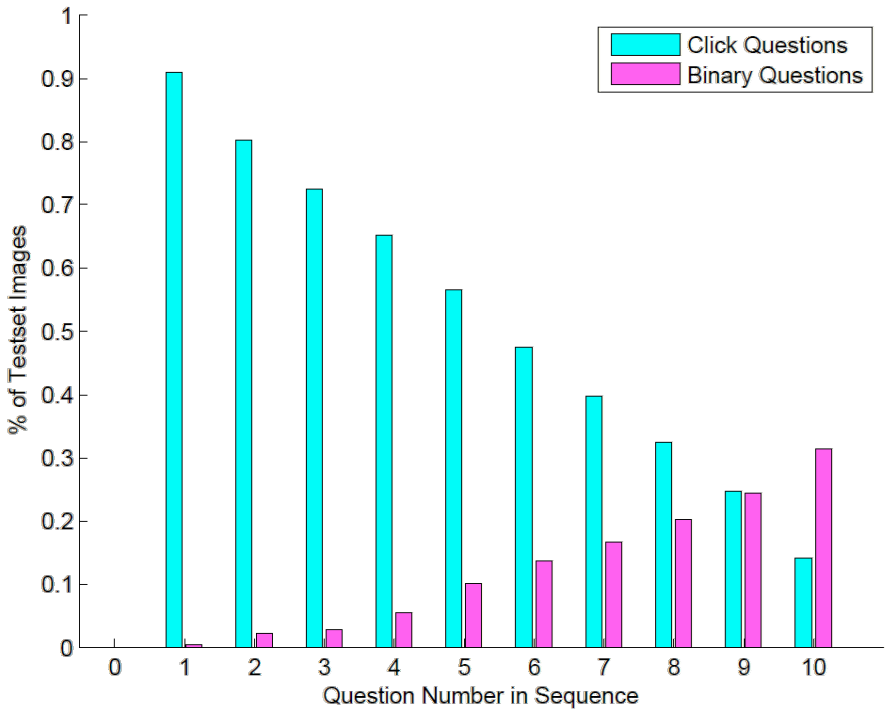
Funding • Powell Foundation Fellowship • NSF Graduate Research Fellowship • Google Focused Research Award

Thanks!

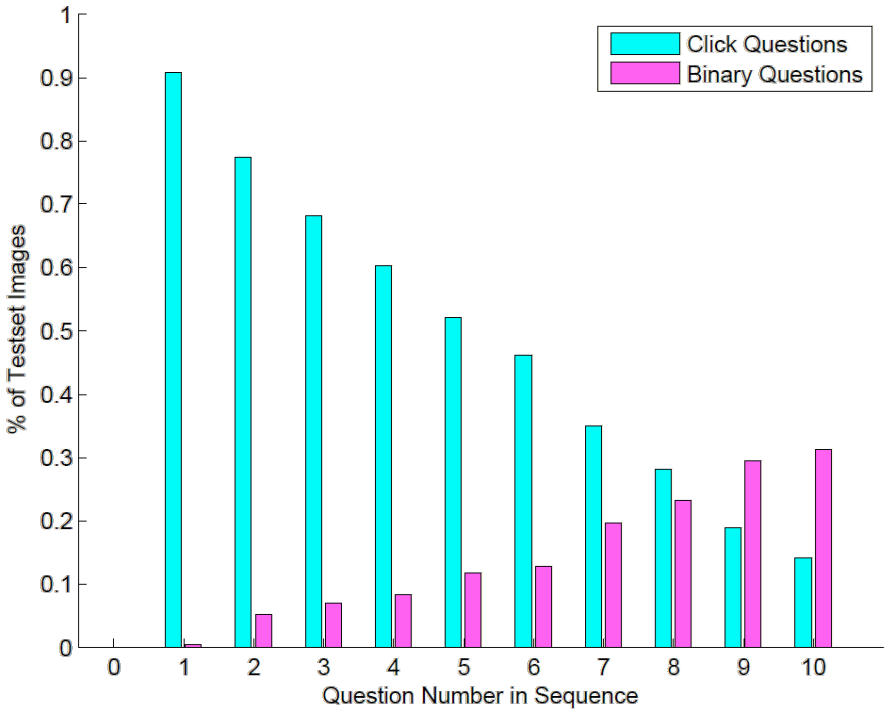
Parts and Attributes

Additional slides

Expected IG Criterion

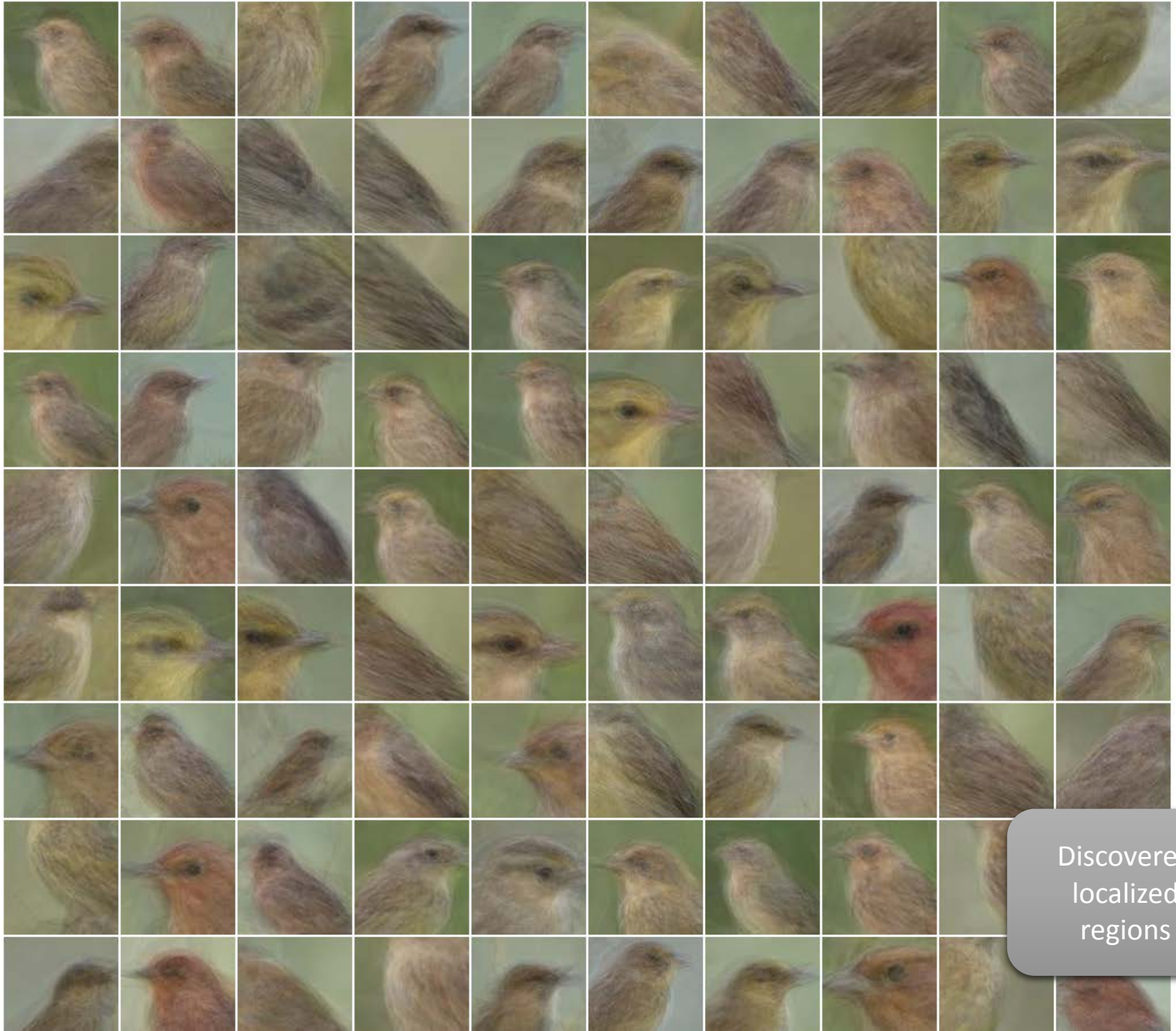


Time Criterion



Similarity Learning

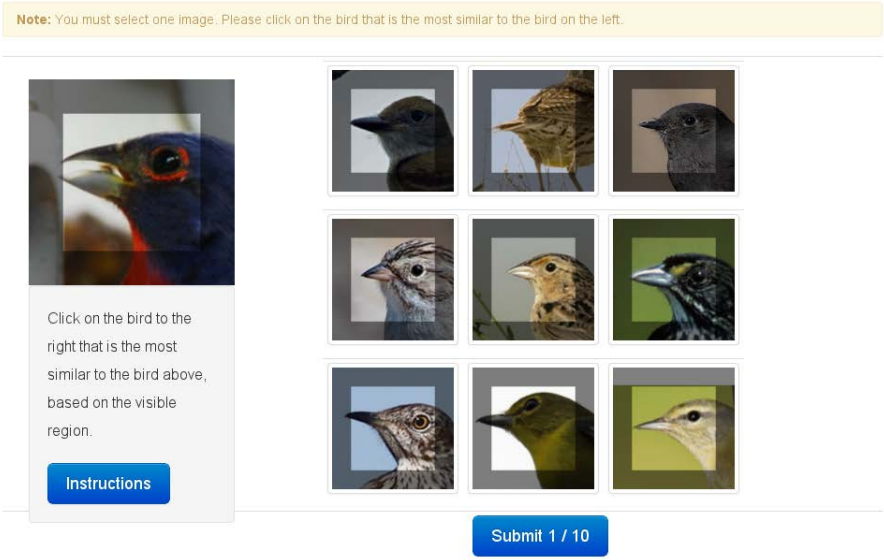
Additional slides



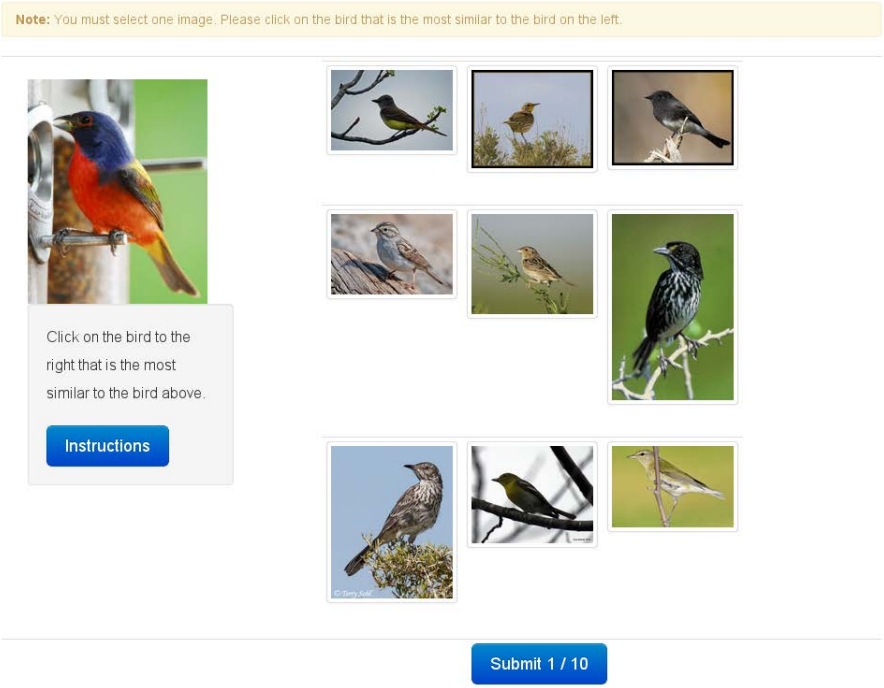
Discovered localized regions

Human perception of similarity

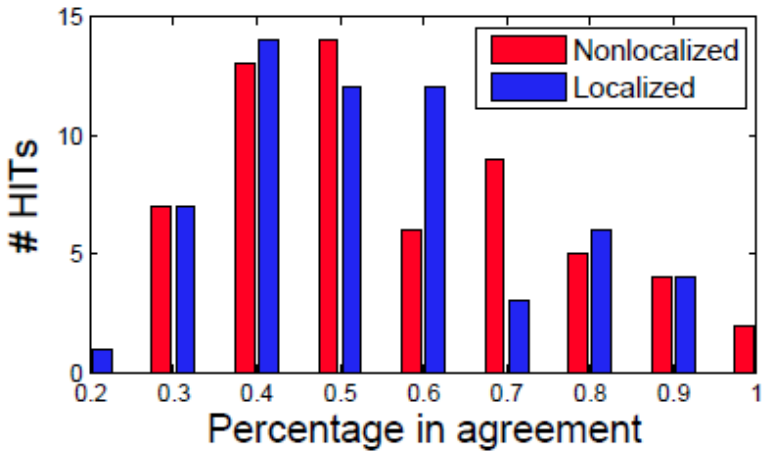
Localized comparisons

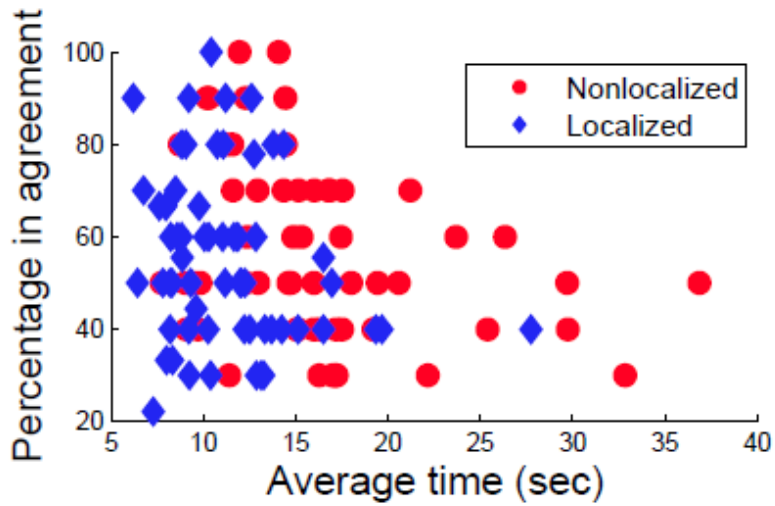


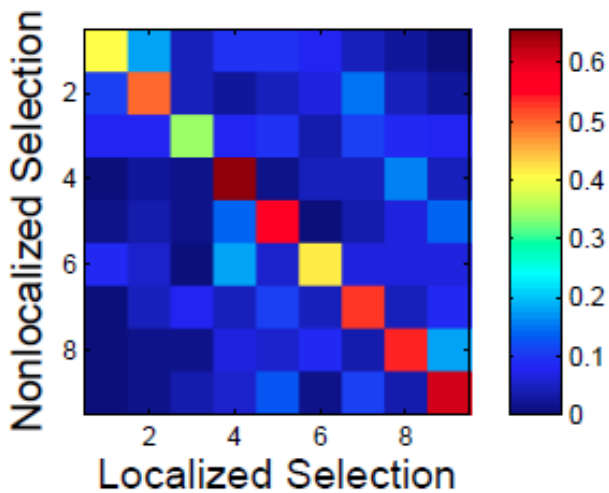
Nonlocalized comparisons



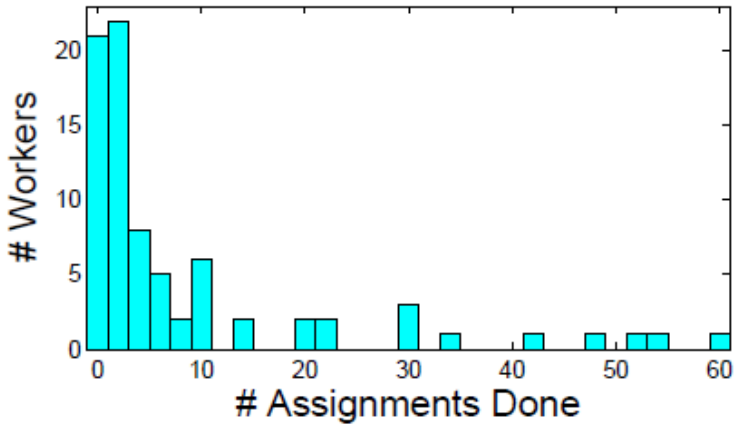
Human perception of similarity

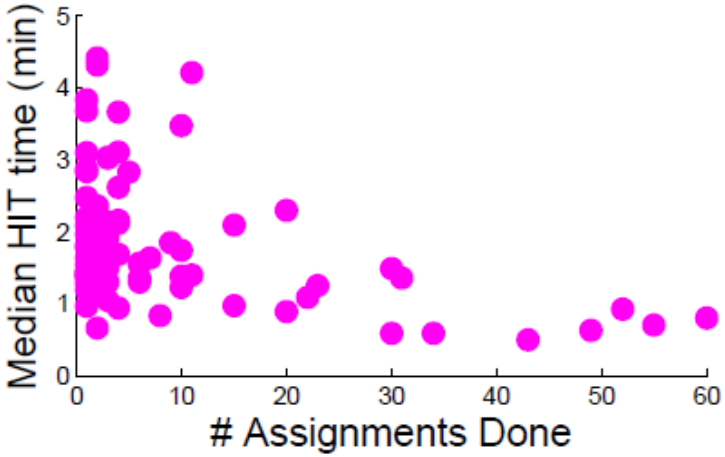


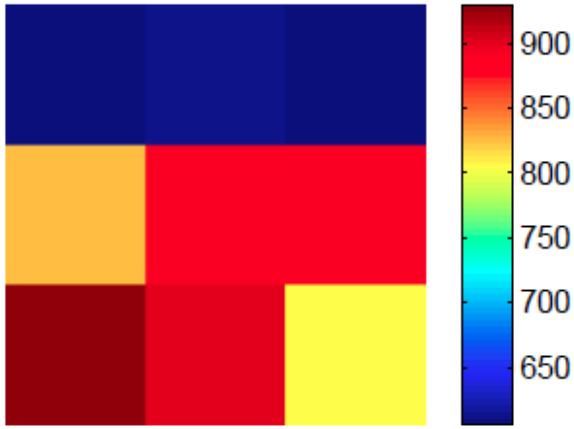


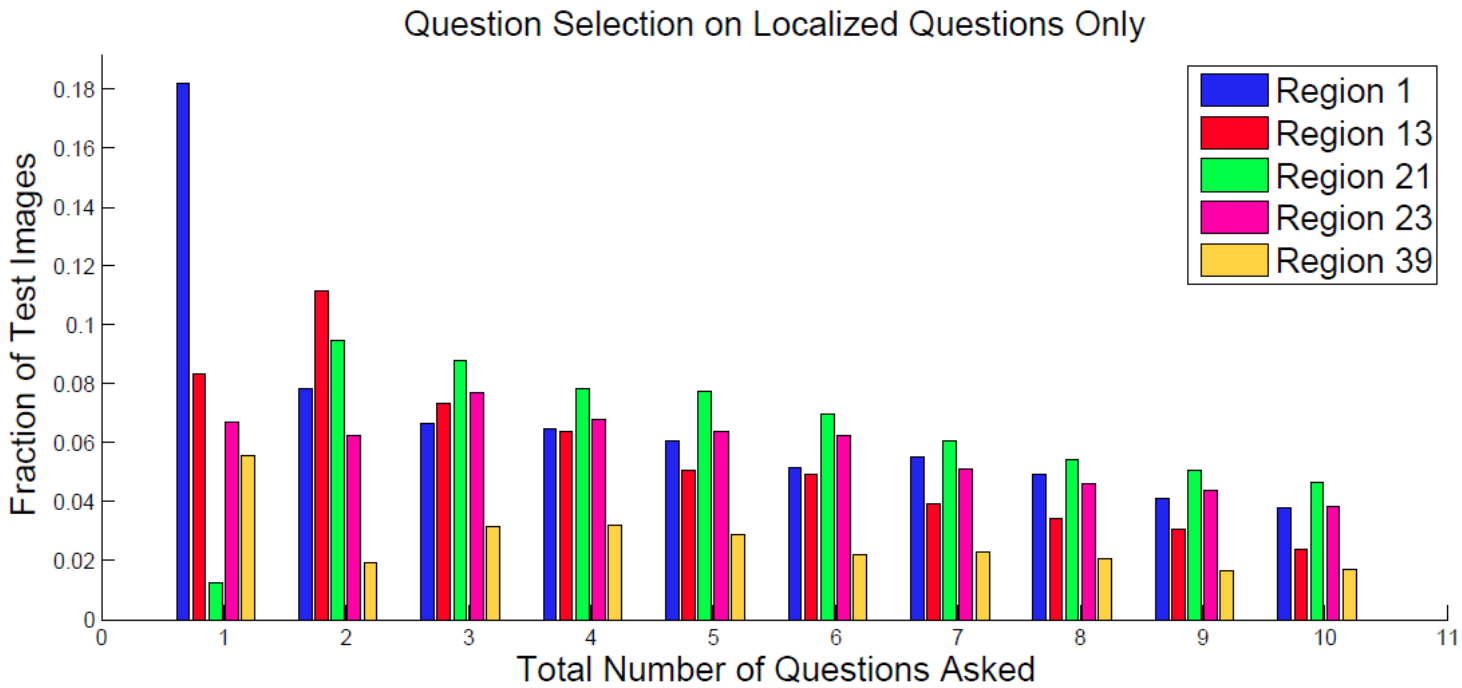


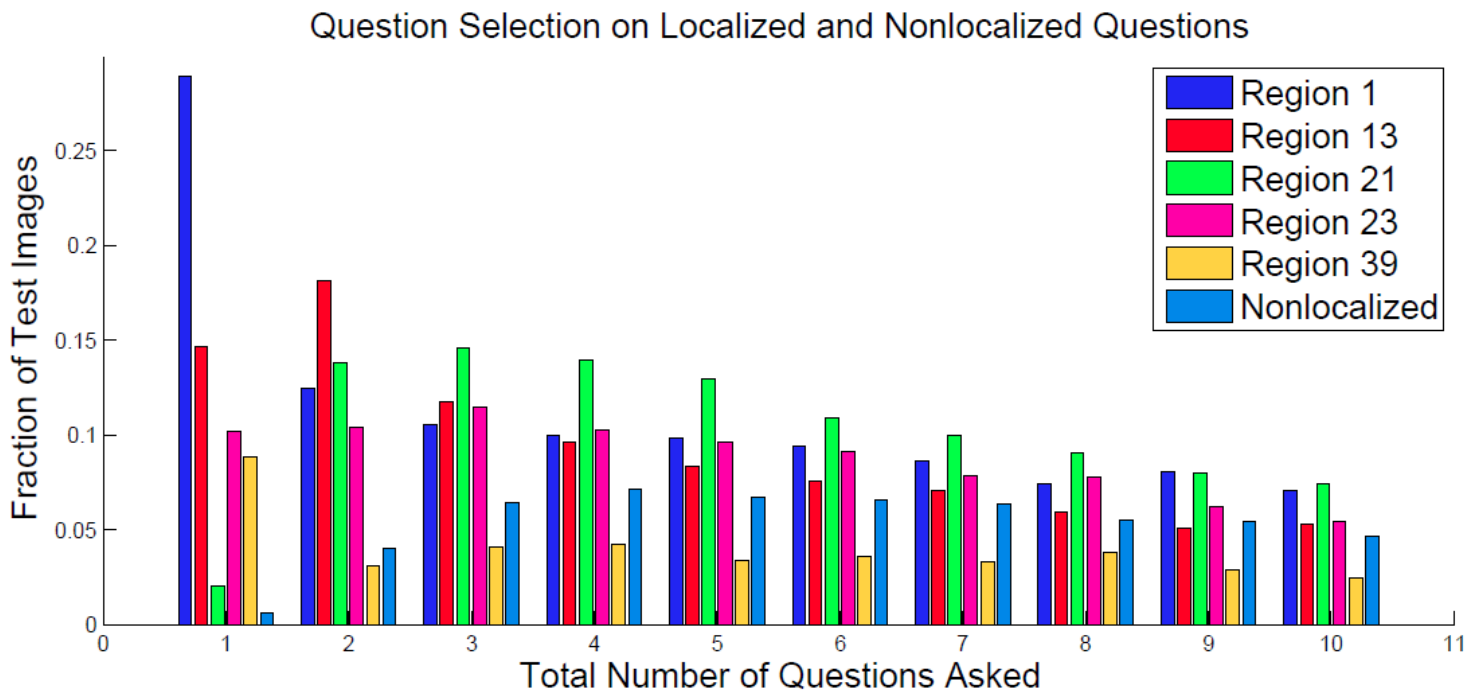
AMT worker behavior








Question Selection on Localized Questions Only
Fraction of Test Images
Total Number of Questions Asked
Region 1
Region 13
Region 21
Region 23
Region 39
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0
1
2
3
4
5
6
7
8
9
10
11


Question Selection on Localized and Nonlocalized Questions
Fraction of Test Images
Total Number of Questions Asked
Region 1
Region 13
Region 21
Region 23
Region 39
Nonlocalized
0
0.05
0.1
0.15
0.2
0.25
0
1
2
3
4
5
6
7
8
9
10
11

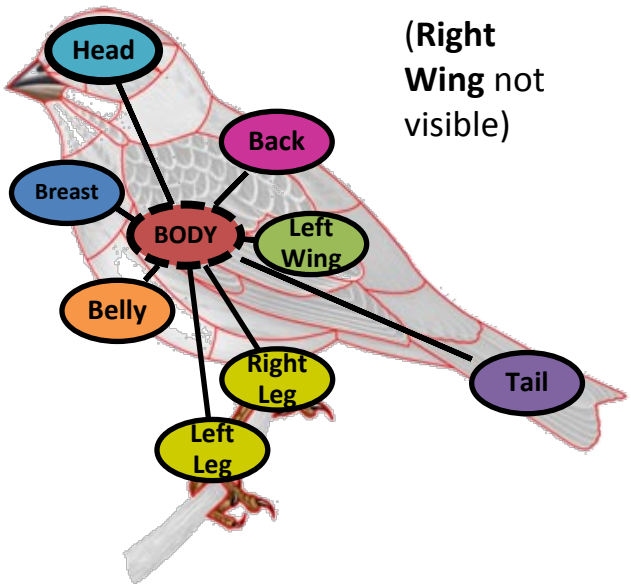
CUB-200-2011 Dataset

CUB-200-2011 dataset

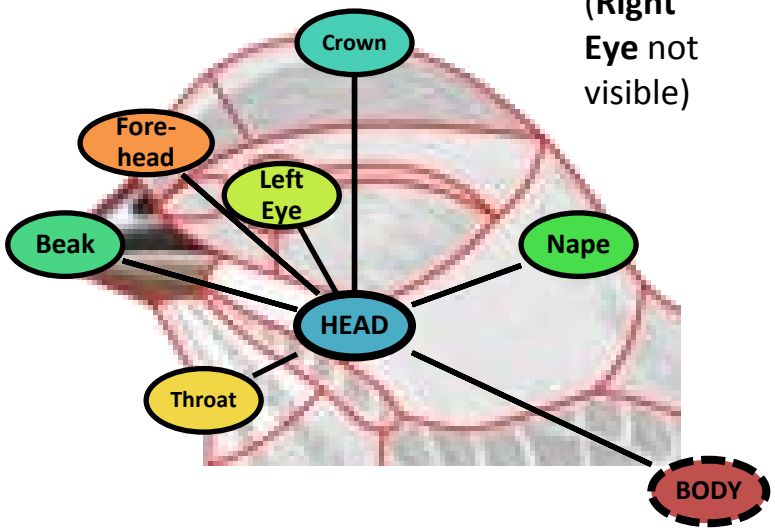
- 200 classes, 11,800 images
- Flickr + MTurk
  - Bird absence/presence
  - Bounding boxes
  - Attribute labeling
  - Part labeling
- MTurk interface
  - Attributes (1 worker)
    - {*not visible, guessing, probably, definitely*}
  - Parts (5 workers)
    - *(x,y)* coordinates
    - Presence

amazonmechanical turk
Artificial Artificial Intelligence

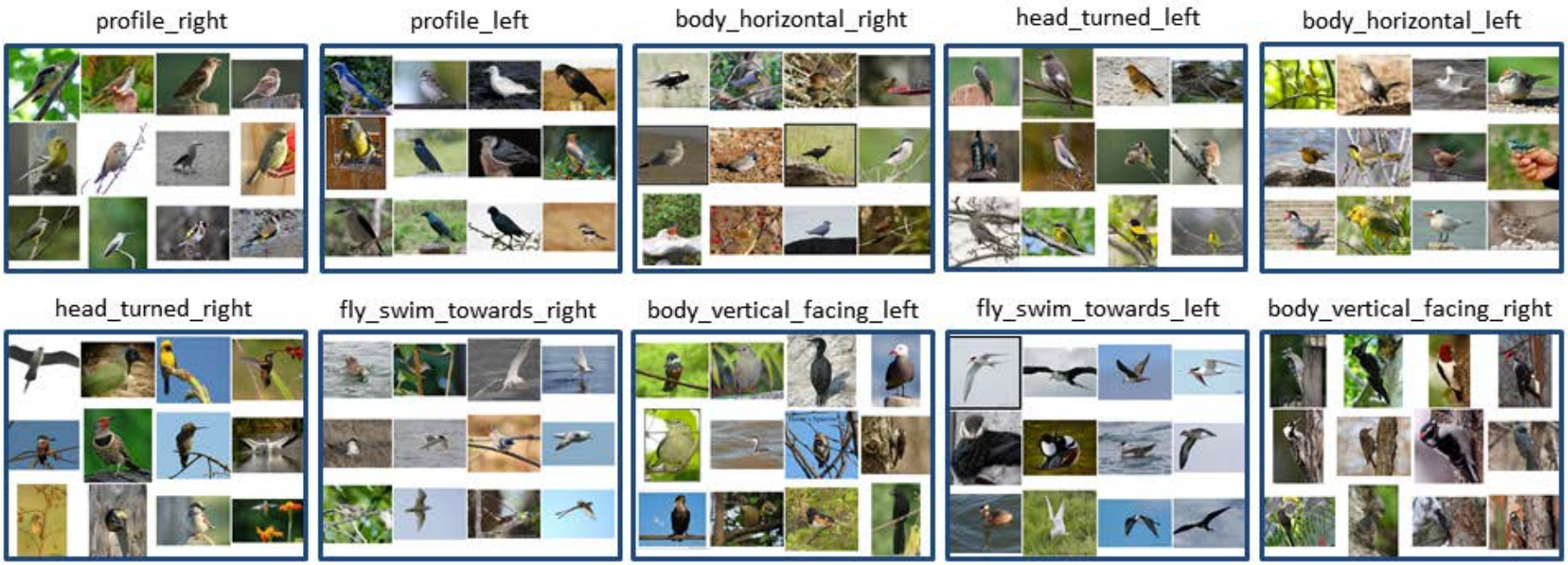
[Wah *et al*. 2011]



(**Right Wing** not visible)



(**Right Eye** not visible)

Part	Attributes	Part	Attributes	Part	Attributes
Beak	*HasBillShape, HasBillColor, HasBillLength*	Back	*HasBackColor, HasBackPattern*	Breast	*HasBreastPattern, HasBreastColor*
Belly	*HasBellyPattern, HasBellyColor*	Fore-head	*HasForeheadColor*	Bird (all parts)	*HasSize, HasShape*
Throat	*HasThroatColor*	Nape	*HasNapeColor*	Head	*HasHeadPattern*
Crown	*HasCrownColor*	Eye	*HasEyeColor*	Leg	*HasLegColor*
Tail	*HasUpperTailColor, HasUnderTailColor, HasTailPattern, HasTailShape*	Wing	*HasWingPattern, HasWingColor, HasWingShape*	Body	*HasUnderpartsColor, HasUpperPartsColor, HasPrimaryColor*

Pose clusters