Peekaboom: A game for locating objects in images

Luis von Ahn, Ruoran Liu and Manuel Blum

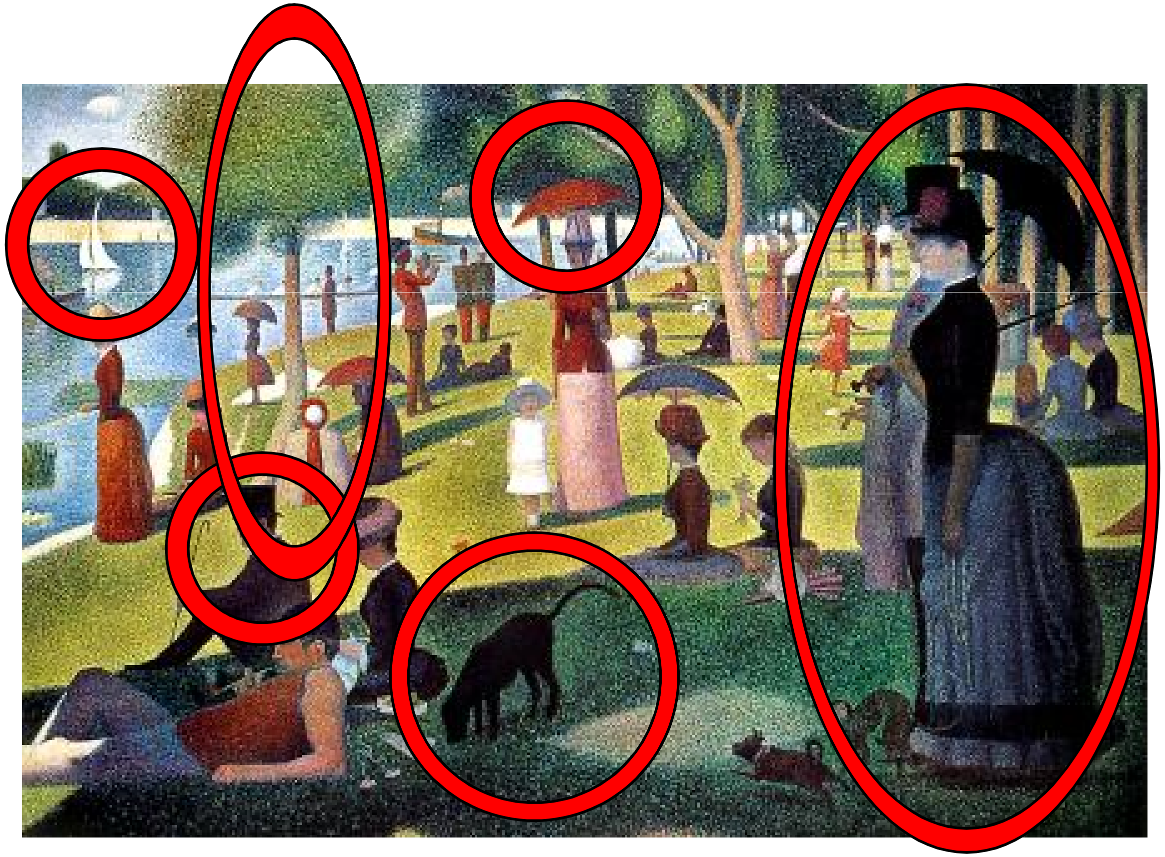
Computer Science Department, Carnegie Mellon University

Presented by: Nils Murrugarra

University of Pittsburgh

Object Location in Images

Given an image, determine what **objects** are present in the image and **locate** them:



Woman

Man

Umbrella

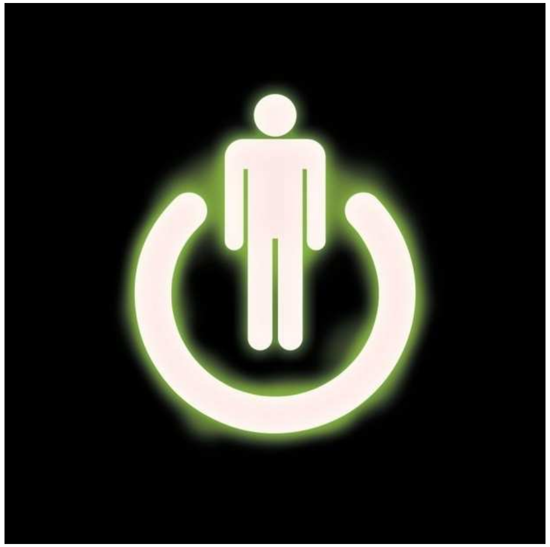
Tree

Sailboat

Dog

Let’s use Human Power

- “Math is hard. Let’s go shopping!” –Barbie
- On similar line of thinking:
  - Programming computers to locate objects in images is hard, so…
  - Let’s not think about that.
  - Instead, humans can do the work for us?



Problems

- **Wait!** Human probably wants:
  - **Enjoyment** – they want to have a good time
  - **Incentives** – they want something in return

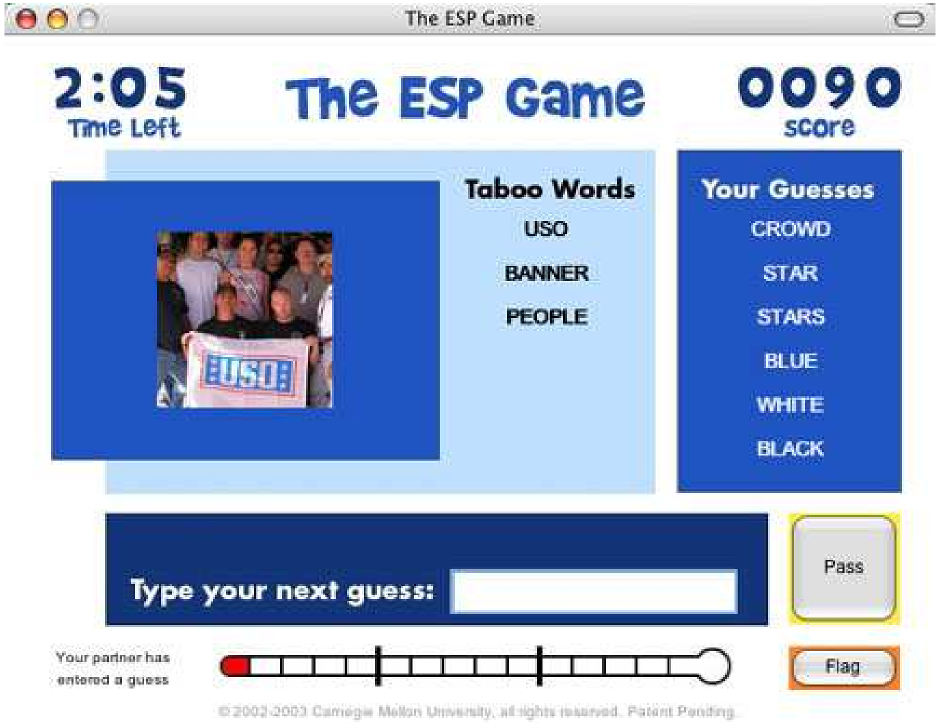
- How to address them?

A Game

- People can do the work for us by playing a game.
- Many questions appears:
  - What will be the **core idea** of the game?
  - How do we **collect data**?
  - How do we ensure the **quality of the data**?

An Earlier Idea:
Luis von Ahn's ESP Game – *Core Idea*

- Two players without communication watch a particular **image**, each one **tries to guess what the other is thinking** about the image.
- If they agree on a word, the game moves on and increases both players' scores.



Taboo Words
USO
BANNER
PEOPLE

Your Guesses
CROWD
STAR
STARS
BLUE
WHITE
BLACK

Type your next guess:

Your partner has entered a guess

© 2002-2003 Carnegie Mellon University, all rights reserved. Patent Pending

A Sample Run

Player 1 Guesses

- Pants
- Model
- Lady

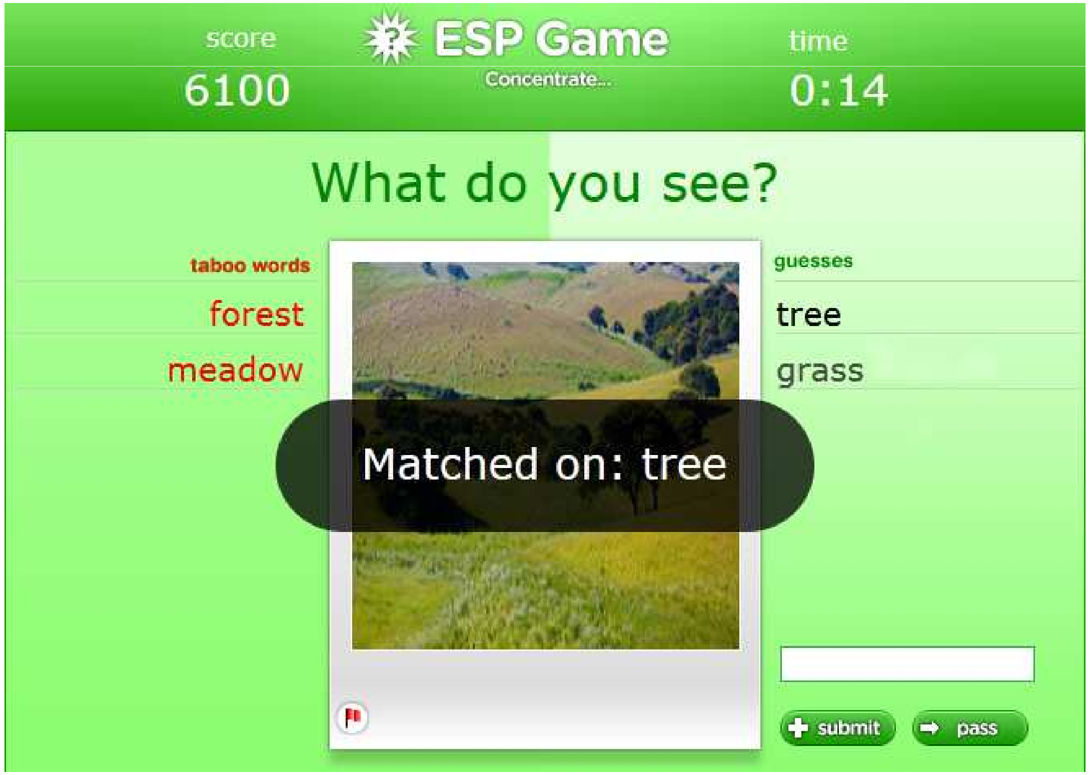
Player 2 Guesses

- Woman
- Shirt
- Girl
- Model

Server: Agreed, “Model”

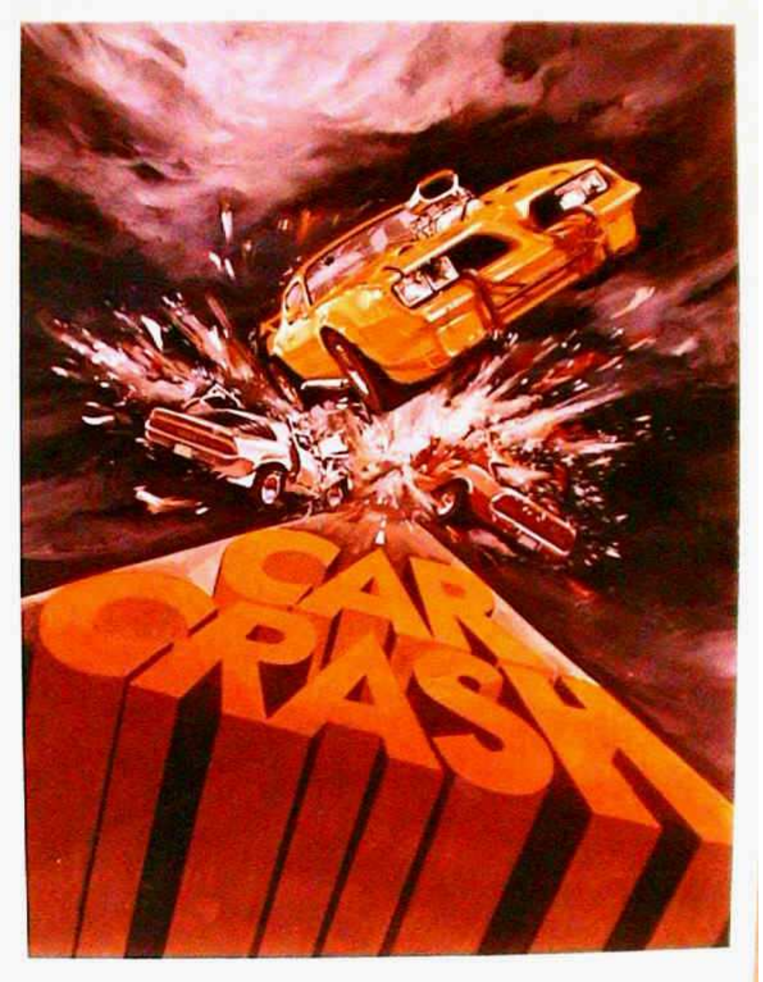
Why ESP Works – *Data Collection and Quality*

- **When two players agrees:**
  - Say **what it is** – In other words this is a “**label**” to the shown image.
  - The fact that two players agree on a label means that this label has a **high quality**.

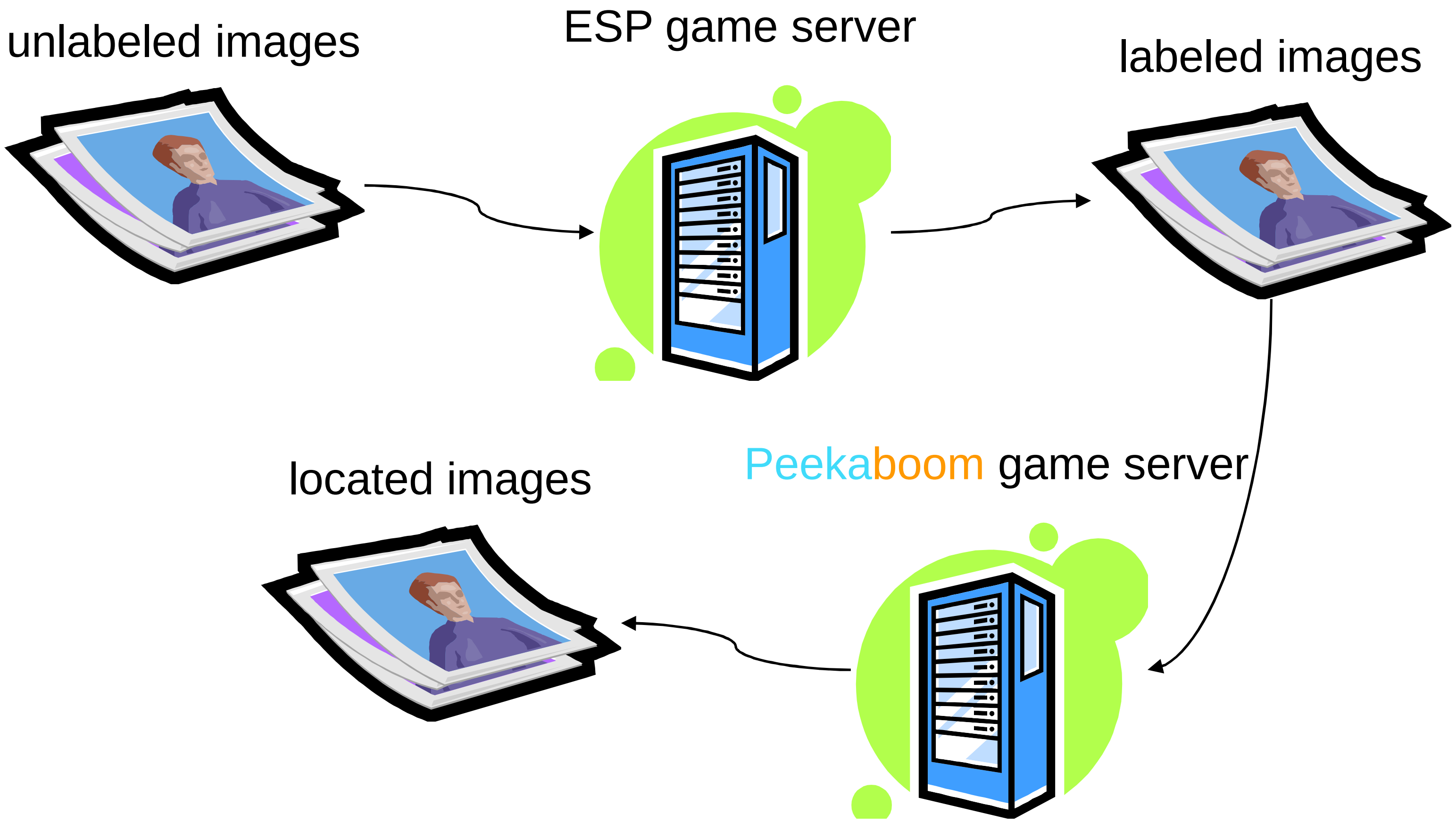


Limitations of ESP

- The ESP Game can label images (**what's in them**), but it **cannot**:
  - **Where** the objects are?.
  - Determine the **way** in which the object appears – does the label "car" refer to the text "car" or an actual car in the image?



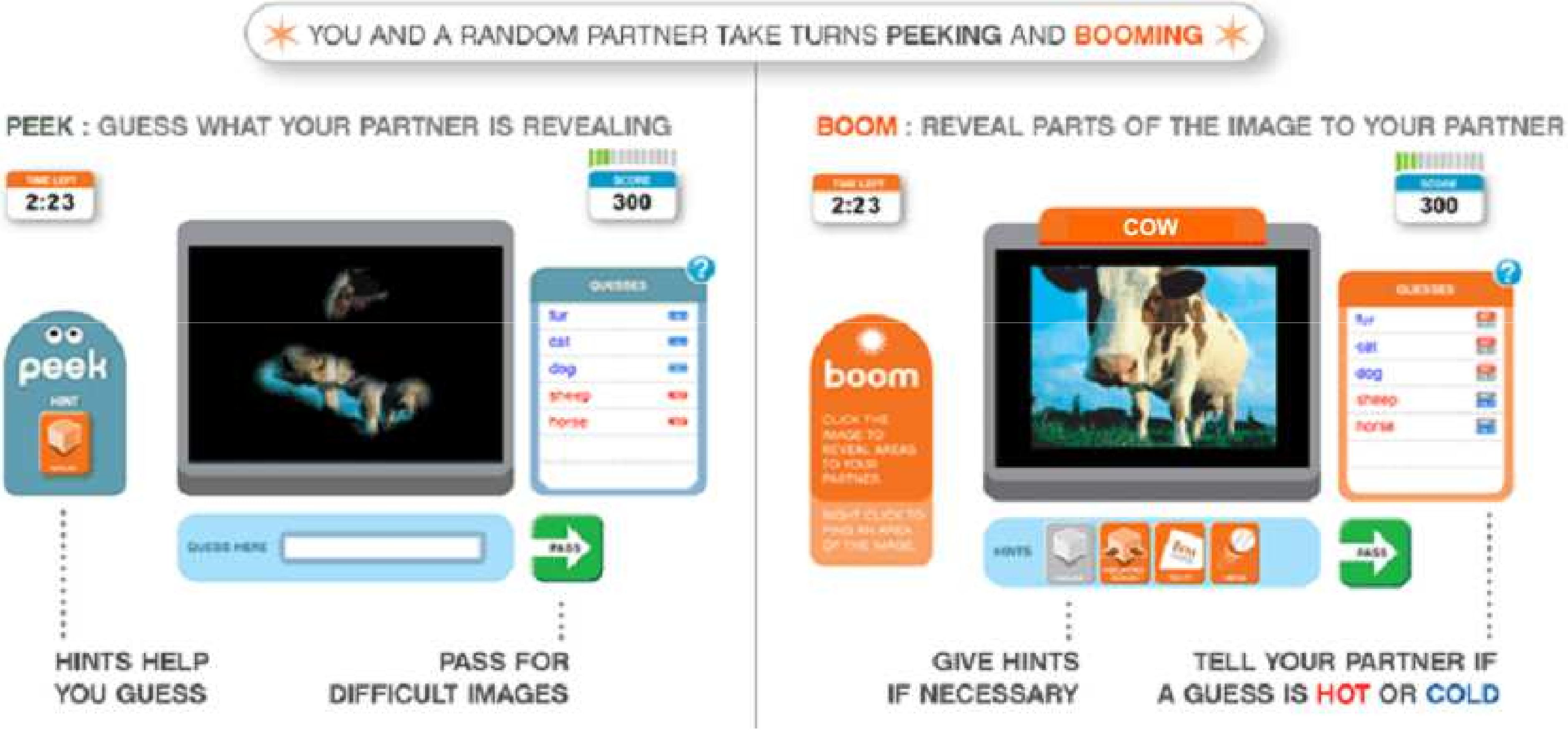
Completing the Image Cycle



A New Idea: Peekaboom – *Core Idea*

- Two players are assigned the roles of **“revealer”** (**BOOM**) and **“guesser”** (**PEEK**).
- The revealer sees an image with a label. The guesser sees nothing.
- The revealer shows the guesser parts of the image. If the guesser guesses correctly, the game continues with new images.

Peekaboom - Interface


YOU AND A RANDOM PARTNER TAKE TURNS PEEKING AND BOOMING

PEEK : GUESS WHAT YOUR PARTNER IS REVEALING

BOOM : REVEAL PARTS OF THE IMAGE TO YOUR PARTNER

HINTS HELP YOU GUESS

PASS FOR DIFFICULT IMAGES

GIVE HINTS IF NECESSARY

TELL YOUR PARTNER IF A GUESS IS HOT OR COLD

TIME LEFT
2:23
SCORE
300
peek
HINT
GUESSES
fur
cat
dog
sheep
horse
GUESS HERE
PASS
COW
boom
CLICK THE IMAGE TO REVEAL AREAS TO YOUR PARTNER
RIGHT CLICK TO PING AN AREA OF THE IMAGE
HINTS
PASS

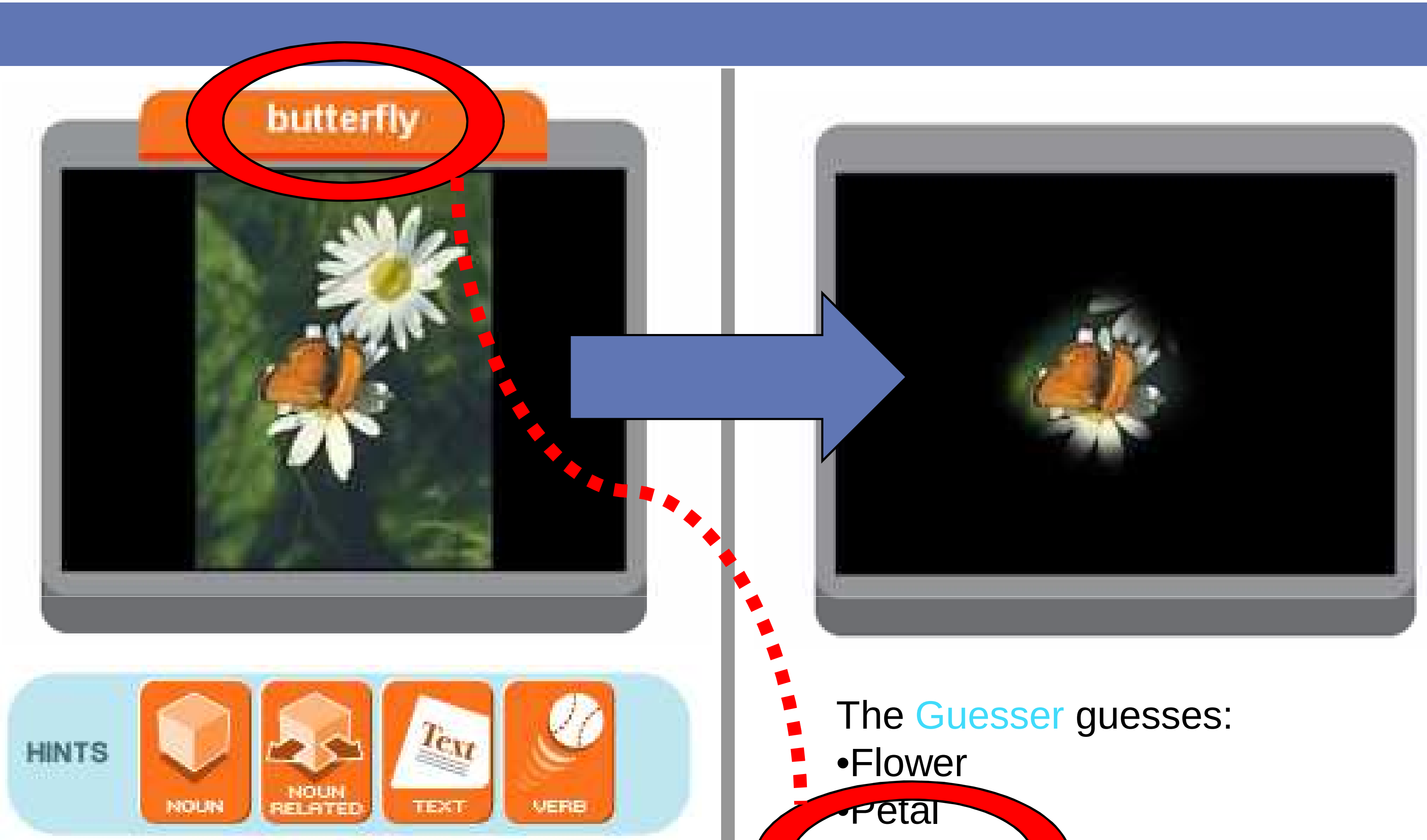
Peek - Guesser

Boom - Revealer

Statement of Purpose

- The authors would like to **collect data** of a lot images **automatically**
- The authors hope that these **data** can be used to train **computer vision** algorithms.

Let’s do an example …



The Revealer clicks on parts of the image and shows them to the Guesser.

The Guesser guesses:

- Flower
- Petal
- Butterfly

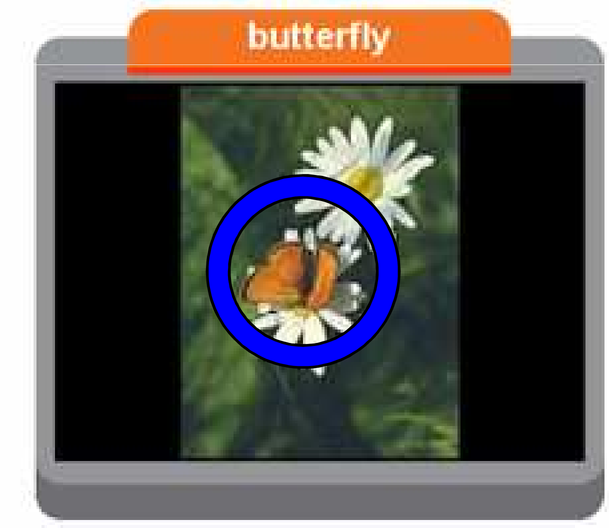
Server: Correct, Butterfly

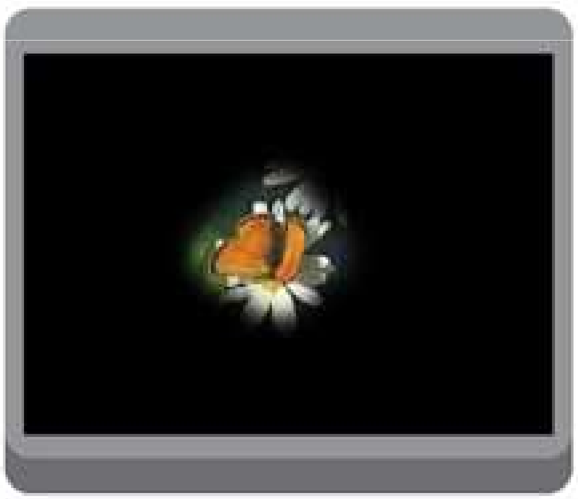
Let’s Play …

https://www.youtube.com/watch?v=tx082gDwGcM&feature=youtu.be&t=1683

Why Peekaboom Works

- To **help** as much as possible the guesser to **guess correctly**, the revealer locates **relevant parts** of the object in the image:

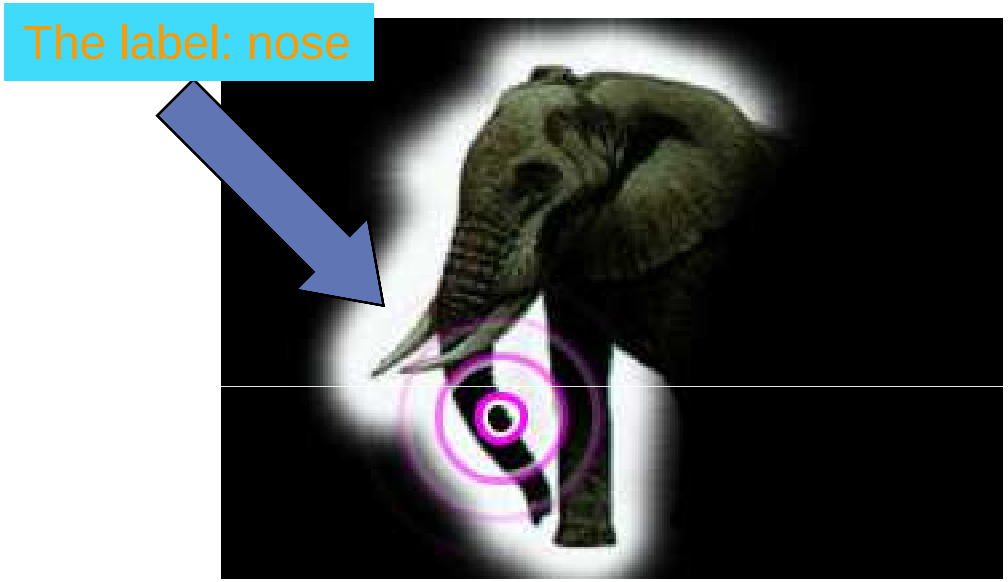




But Wait, There’s More

- Peekaboom **not only locates objects**:
  - It gives the **context** necessary to identify them.
  - It Classifies the image as “Text”, “Noun”, or “Verb” using the **hints** option.
- Let’s learn more about these functionalities

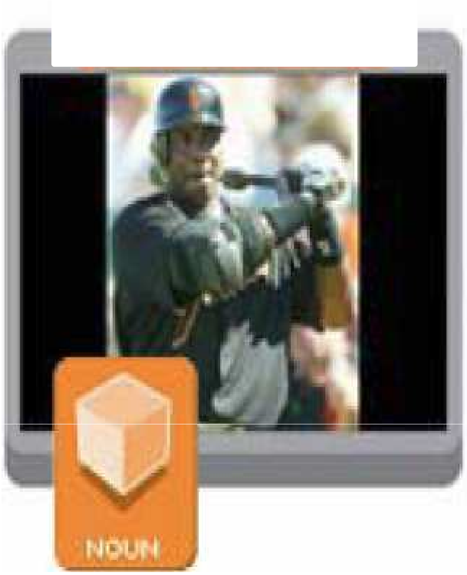
Object Context

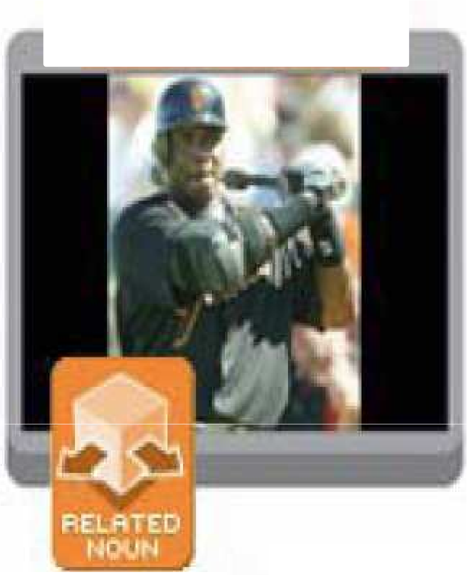


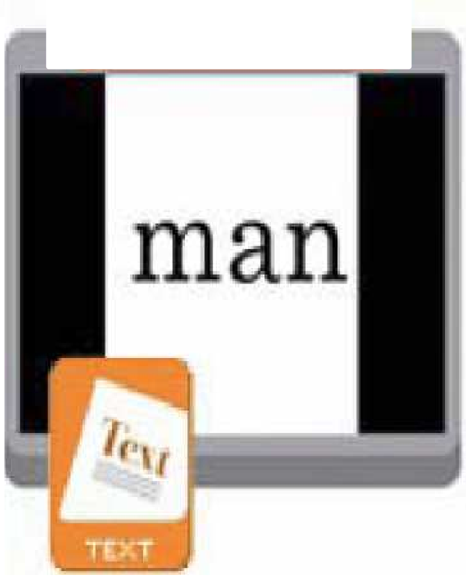
- **Pings** help separate the **context** of object with the object itself.

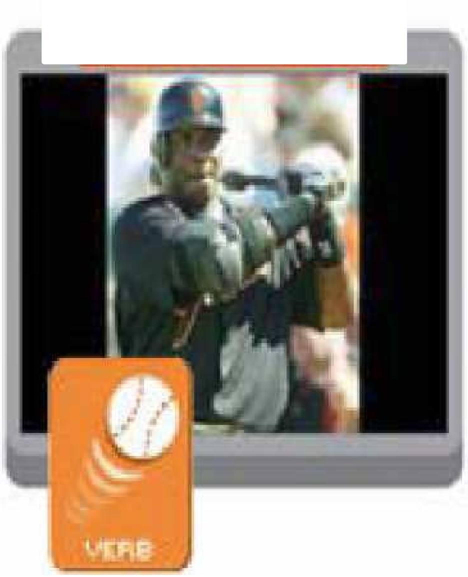
- They **help** the guesser **distinguish** nose from other possibly correct labels like “elephant” and “ear”.

Hints

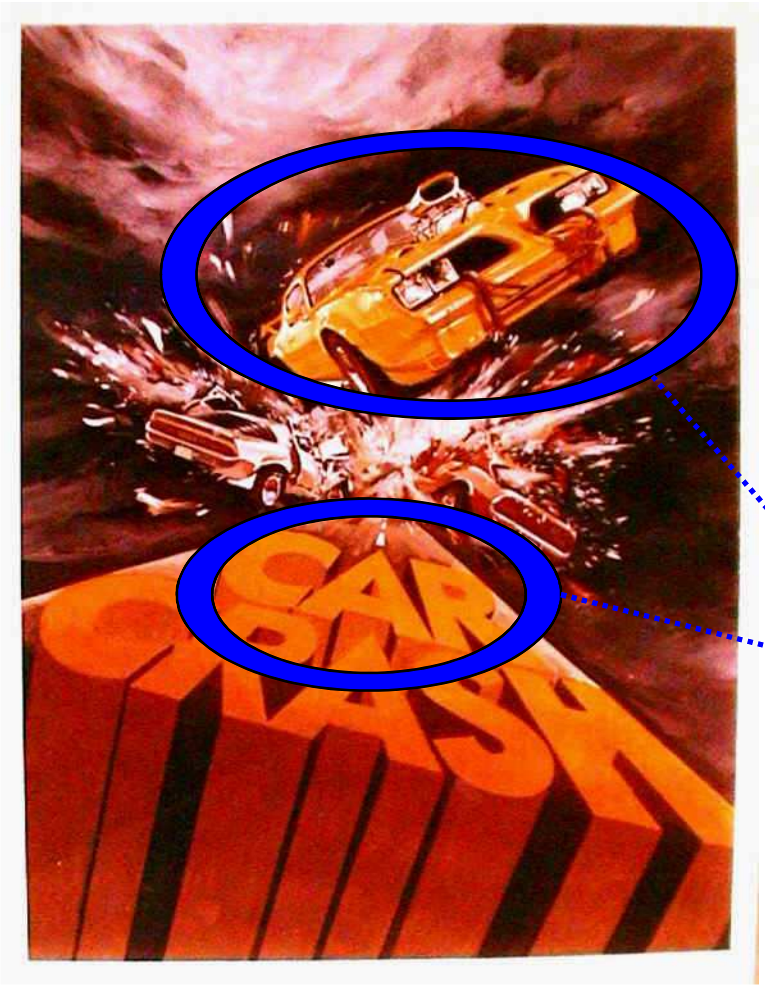






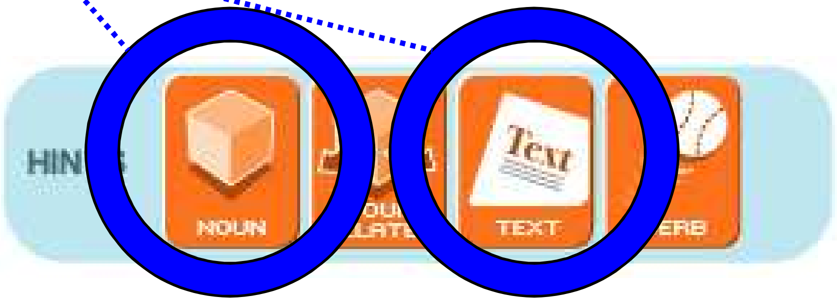


The Role of Hints



The label “car” is ambiguous --

The hints help distinguish the manner in which the label “car” appears:
this is “car”
this is the object “car”
this is also “car”
this is the text “car”



How to involve more participants in the game?

Game Points

Game Points

- Peek guesses the correct word (+ 50)
- Points are not subtracted for passing (+ 0)
- Peek guesses the correct word and Boom had used a hint (+ 25 extra)
- Points are not given for usage of the hot/cold buttons (+ 0)

Bonus Points

- Obtain up to get + 150 points
- Points depend on how far one participant's click is from his/her partner's corresponding click (+ 0 ~ 10)
- If the object are not in the image, players can pass (+25)

Collecting Image Metadata – *Data Collection*

- Data from **Area Revealed**: Which **pixels** are necessary to guess the word?
- Data from **hints**: what is the **relation** between word and image?
- Data from **pings**: which **pixels** are **inside** the object?
- Data from **sequence of Boom's clicks**: What are the most **relevant** aspects of the object?
- Data from **Pass Button**: Elimination of **poor/difficult** image-word pairs

Cheating – *Data Quality*

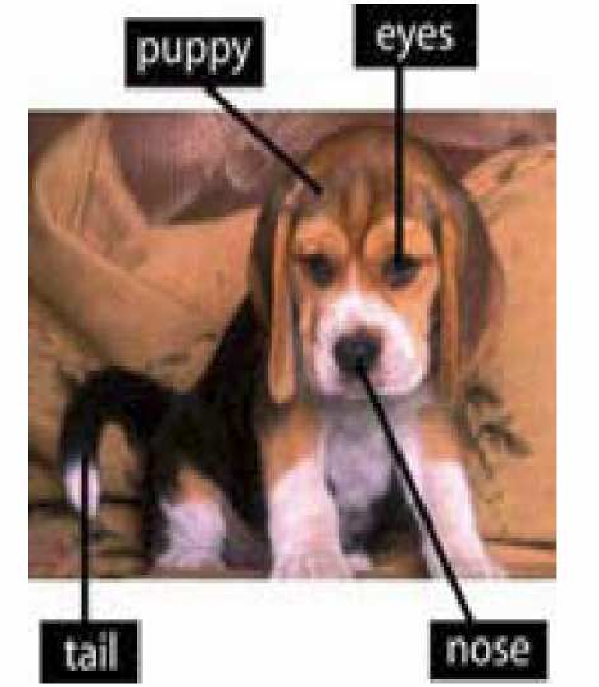
- **Why to worried?** If the two players cheat on the game, the data is not reliable.

Multiple anti-cheating mechanisms

- To avoid match participants that start at the **“same time“** : The **player queue**
- To avoid **geographically proximity**: **IP address** checks
- To avoid **bots**: **Blacklists** after consistent failure on “seed” images
- To avoid **”cheating communication”**: **Limited** freedom to enter guesses

Applications

- Improving Image-Search Results
- Object Bounding-Boxes
  1. Given an image, create a **matrix of 0's**
  2. For each **click** in its surrounding area (radius 20 pixels). Add **+1** to the matrix position
  3. **Combine** different games for the same image-word pair.
  4. Apply a **threshold** of 2 (at least 2 players agree)
  5. **Cluster** the pixels to get bounding boxes
- Using Ping Data for Pointing
  - Select a random ping



Evaluation

Is this an effective way to collect data?

Yes!

Game is enjoyable

- Each person played average of **158.7 images**
- That’s **72 96 minutes** per person in one **month**!
- **User reviews**

Usage Statistics

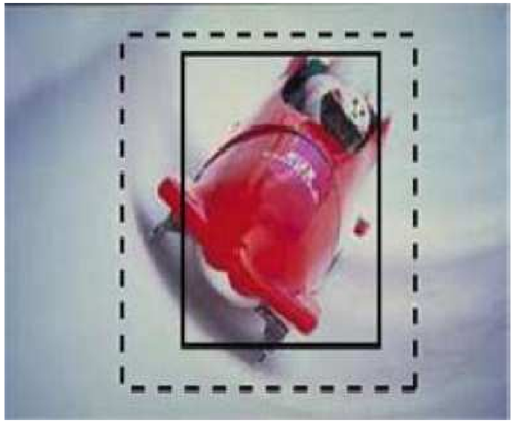
- August 1, 2005 ~ September 1, 2005
- 14153 people and 1122998 pieces of data

Evaluation: Accuracy of Collected Data

Accuracy of Bounding Boxes

Are they good compared to bounding boxes collected in a non-game setup?

- It was performed in 50 image-word (nouns) pairs
- Given a word, four volunteers were asked to draw a bounding box around the object that the word refers to.
- **Average overlap:** 0.754
- **Standard deviation:** 0.109



$$\mathrm{OVERLAP}(A,B) = \mathrm{AREA}(A \cap B) \,/\, \mathrm{AREA}(A \cup B)$$

Accuracy of Pings

- It was verified if the Peekaboom object pointers are indeed inside the objects
- Given a pointer, three volunteer determine if it is inside the object or not.
- **100%** of the pointers were inside the object referred by the word

Discussion

- What are some disadvantages/weaknesses of Peekaboom?
- Can you think of any other applications of Peekaboom?

Conclusion

❖Peekaboom is an enjoyable game to collect image data achieving :

- **Low costs** – One game server.
- **Data with Good Quality**– Accurately locate objects in images.
- **Large Quantity of data**– Locate objects in millions of images.

Questions

References

[1]. Von Ahn, L., Liu, R., & Blum, M. (2006, April). Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (pp. 55-64). ACM.

[2]. Slides version of " Peekaboom: A game for locating objects in images.“ Source: http://www.slideserve.com/rachel/peekaboom-a-game-for-locating-objects-in-images

[3]. Slides version of " Peekaboom: A game for locating objects in images.“ Source: http://nrl.iis.sinica.edu.tw/Web2.0/presentation/ESP_Game_and_Peekaboom.ppt

[4]. Slides version of " Peekaboom: A game for locating objects in images.“ Source: http://cgit.nutn.edu.tw:8080/cgit/PPTDL/LZJ_800224182928.PDF

[5]. Slides version of " Peekaboom: A game for locating objects in images.“ Source: http://www.eecs.harvard.edu/cs286r/courses/fall08/files/AngelaCS286r.pdf

[6]. Video: Human Computation. Source: https://www.youtube.com/watch?v=tx082gDwGcM

Crowdsourcing Annotations for Visual Object Detection

Hao Su, Jia Deng, Li Fei-Fei

Computer Science Department, Stanford University

Presented by: Nils Murrugarra

University of Pittsburgh

Motivation

Motivation:

- A **large quantity** of **precise bounding boxes** are required to learn good object detectors.

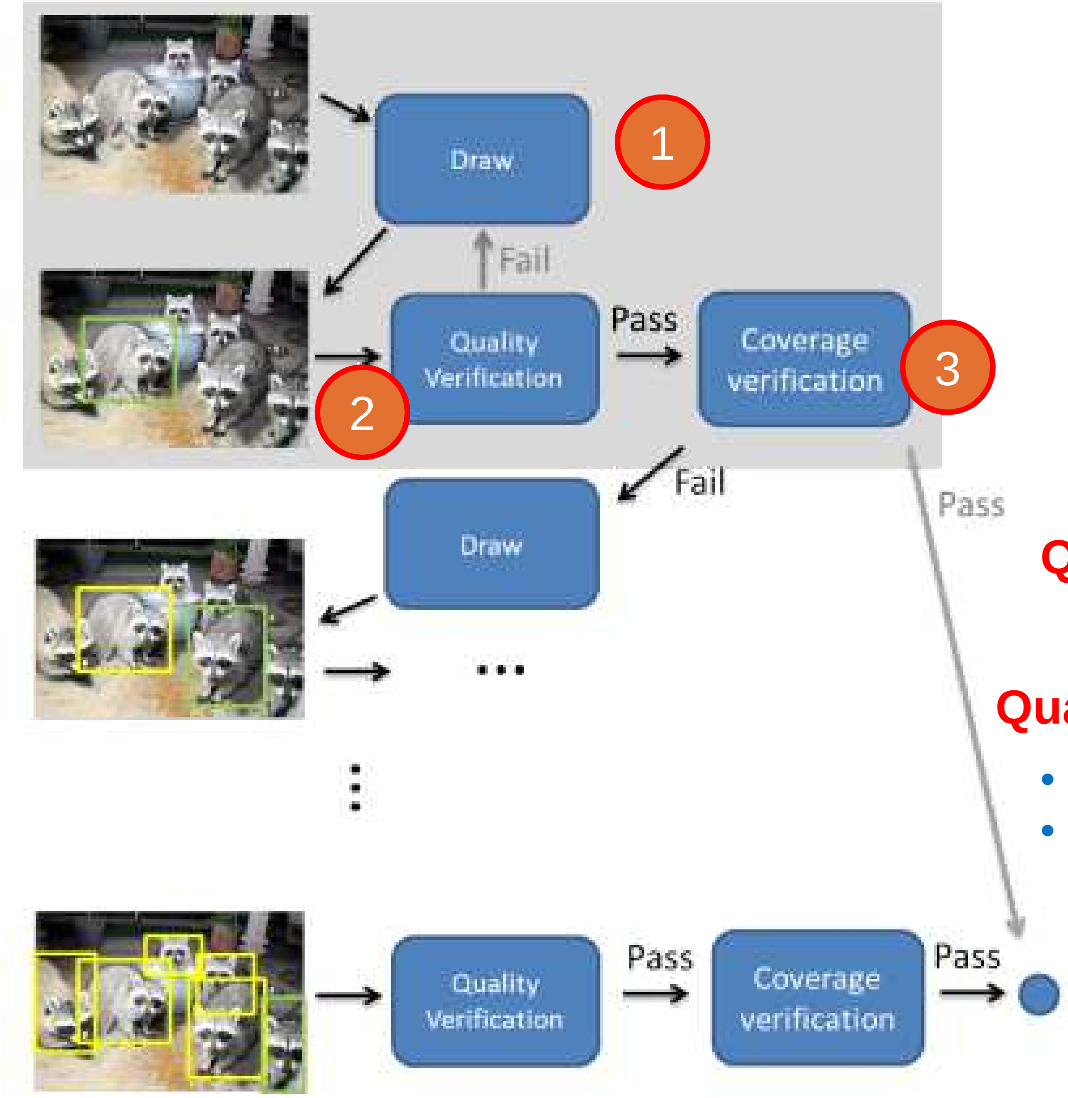
Goal:

- Crowd-source bounding boxes annotations

Challenges:

- Control the data quality with minimal cost.

Method Overview

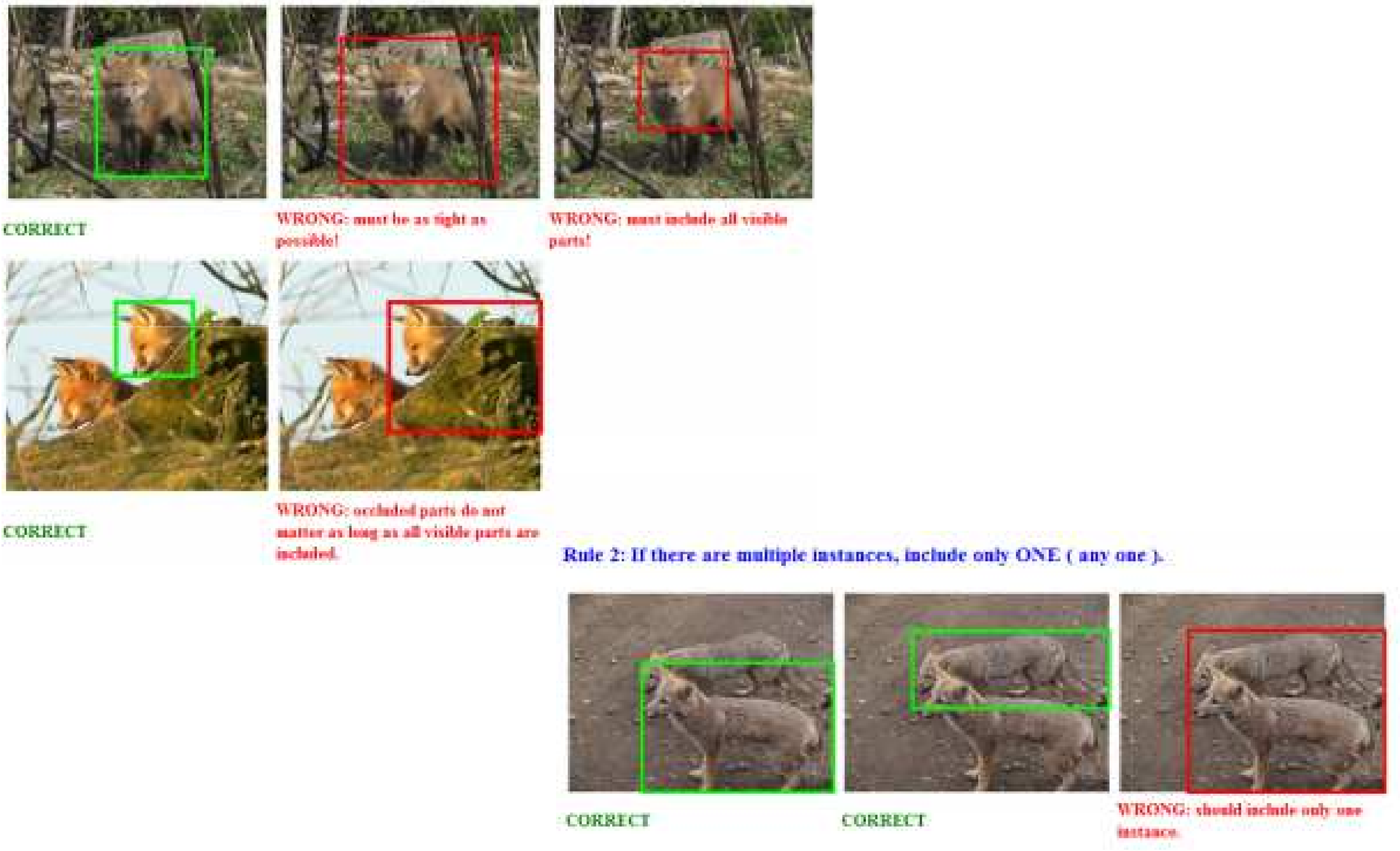


Qualification Test

Qualification Control

- Good Bounding Boxes
- Bad Bounding Boxes

Method – Drawing Task



CORRECT

WRONG: must be as tight as possible!

WRONG: must include all visible parts!

CORRECT

WRONG: occluded parts do not matter as long as all visible parts are included.

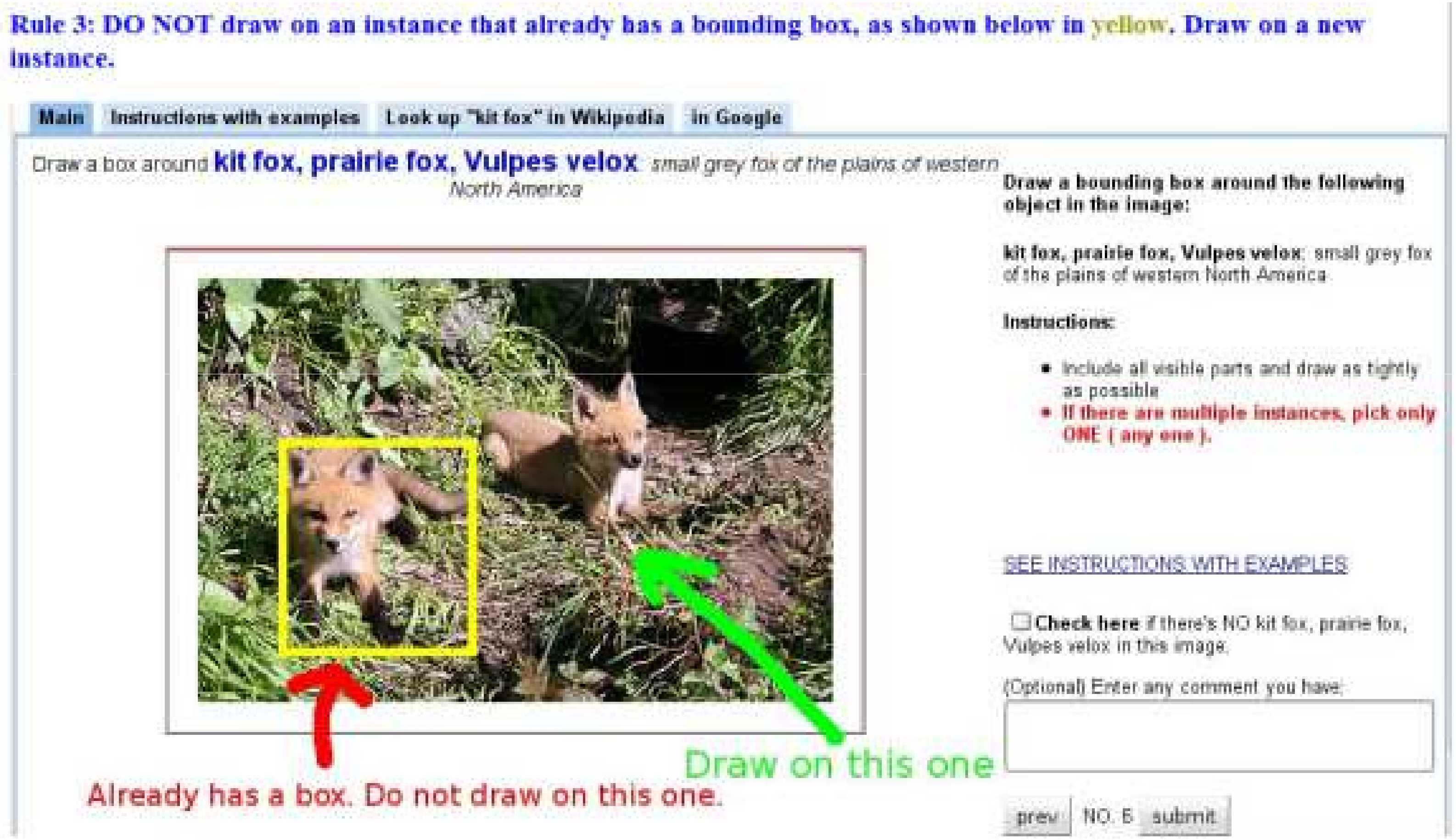
Rule 2: If there are multiple instances, include only ONE (any one).

CORRECT

CORRECT

WRONG: should include only one instance.

Method – Drawing Task



Rule 3: DO NOT draw on an instance that already has a bounding box, as shown below in yellow. Draw on a new instance.

Draw a box around **kit fox, prairie fox, Vulpes velox** *small grey fox of the plains of western North America*

Already has a box. Do not draw on this one.

Draw on this one

Draw a bounding box around the following object in the image:

kit fox, prairie fox, Vulpes velox: small grey fox of the plains of western North America

Instructions:

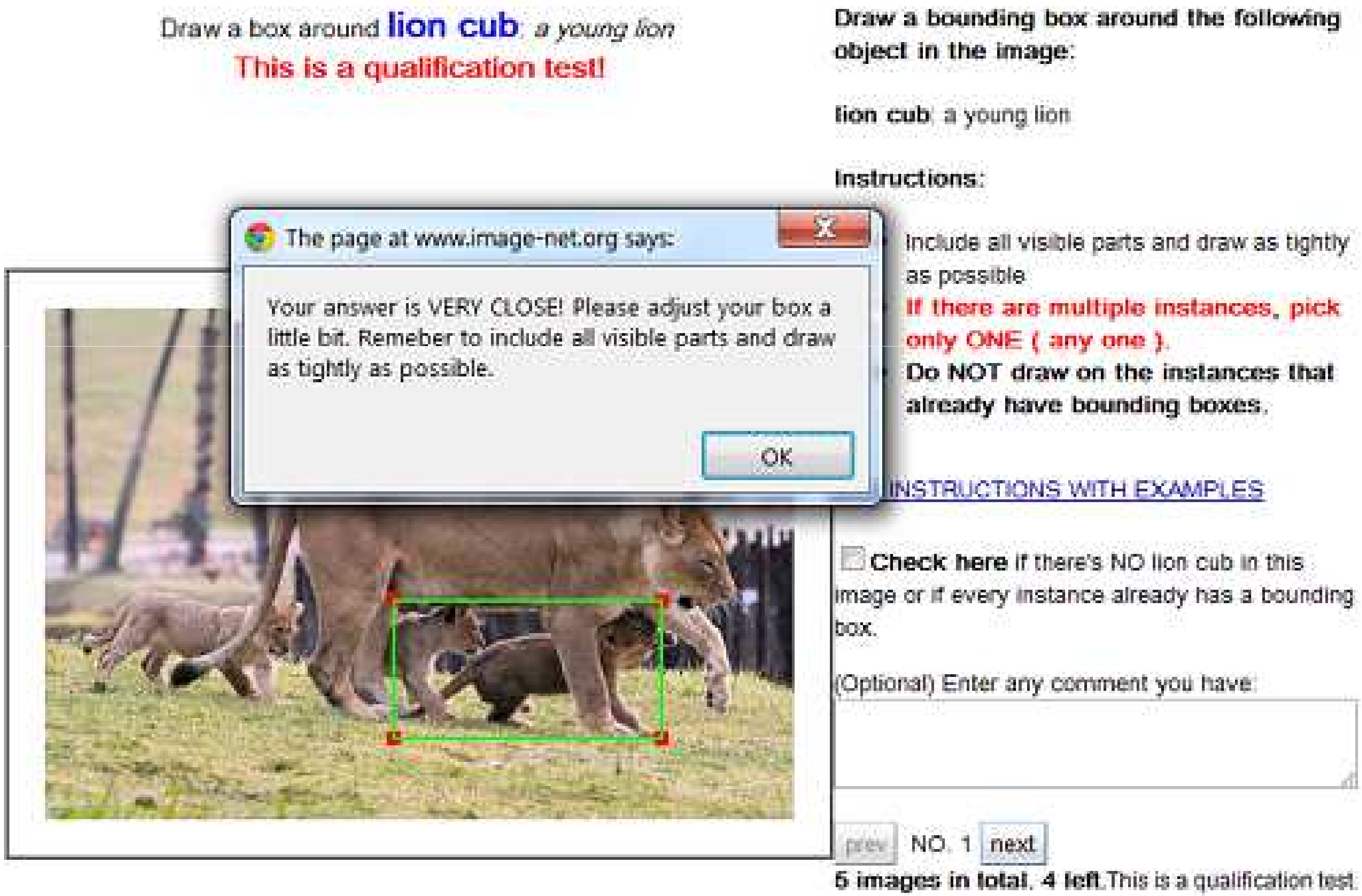
- Include all visible parts and draw as tightly as possible
- **If there are multiple instances, pick only ONE (any one).**

SEE INSTRUCTIONS WITH EXAMPLES

☐ **Check here** if there's NO kit fox, prairie fox, Vulpes velox in this image.

(Optional) Enter any comment you have:

Method – Drawing Task



Draw a box around **lion cub**: *a young lion*

This is a qualification test!

The page at www.image-net.org says:

Your answer is VERY CLOSE! Please adjust your box a little bit. Remeber to include all visible parts and draw as tightly as possible.

Draw a bounding box around the following object in the image:

lion cub: a young lion

Instructions:

- Include all visible parts and draw as tightly as possible
- **If there are multiple instances, pick only ONE (any one).**
- **Do NOT draw on the instances that already have bounding boxes.**

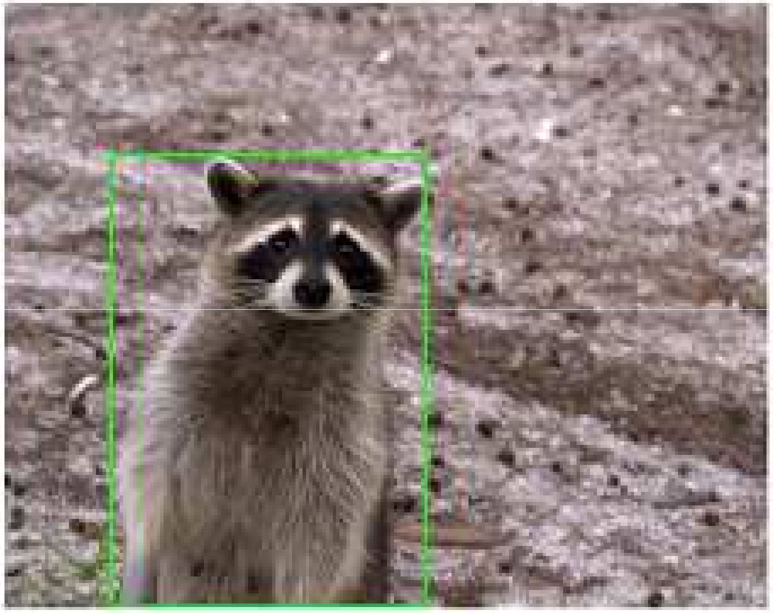
INSTRUCTIONS WITH EXAMPLES

□ **Check here** if there's NO lion cub in this image or if every instance already has a bounding box.

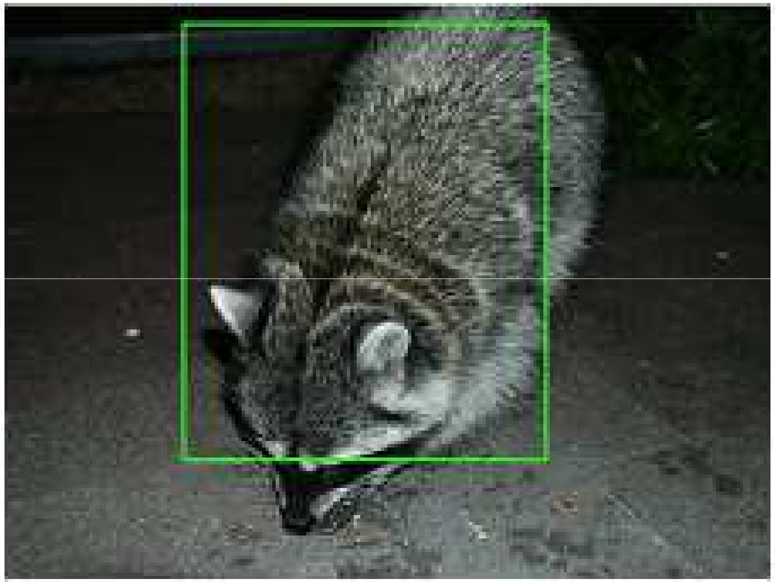
(Optional) Enter any comment you have:

5 images in total, 4 left. This is a qualification test.

Method – Quality Verification Task

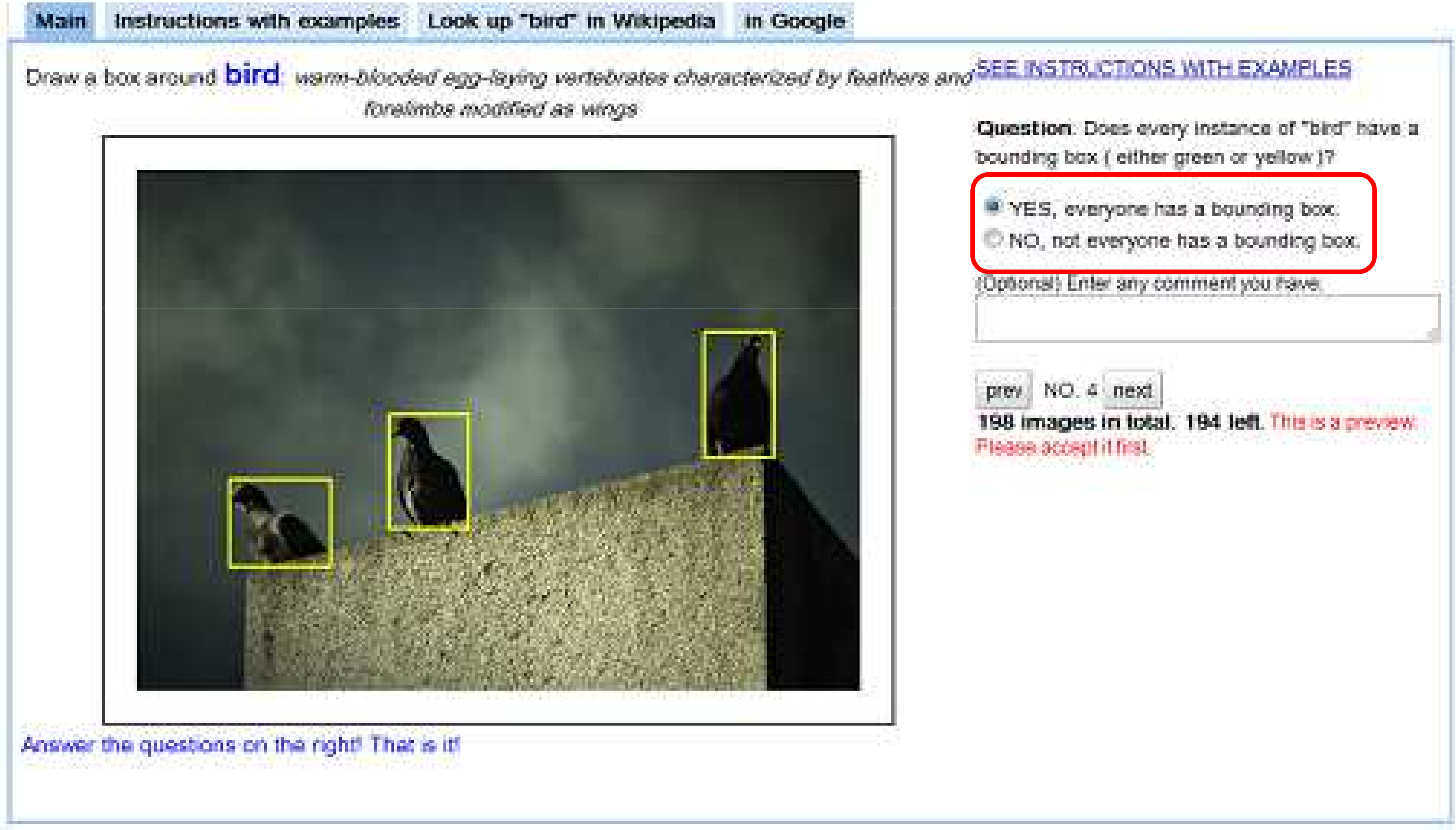


Good Annotation



Bad Annotation

Method – Coverage Verification Task



Draw a box around **bird** *warm-blooded egg-laying vertebrates characterized by feathers and forelimbs modified as wings*

SEE INSTRUCTIONS WITH EXAMPLES

Question: Does every instance of "bird" have a bounding box (either green or yellow)?

● YES, everyone has a bounding box.

○ NO, not everyone has a bounding box.

(Optional) Enter any comment you have.

198 images in total. 194 left. This is a preview. Please accept it first.

Answer the questions on the right! That is it!

Evaluation

Dataset

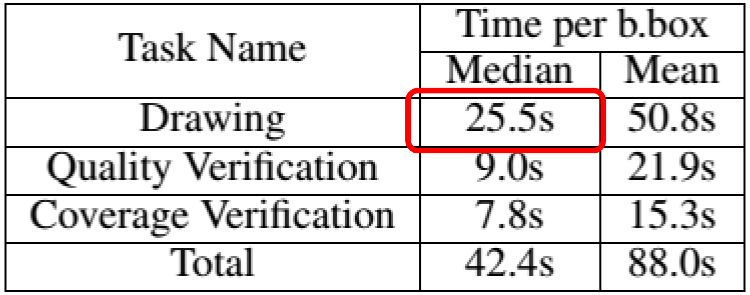
- 200 images were selected over 10 categories on the Imagenet database.

Overall Quality

- It was manually inspected
- **97.9%** of images are **completely covered** with bounding boxes. The remaining 2.1% are difficult cases.
- **99.2%** are **accurate** (tight as possible)

Overall Cost

- The proposed method is **cheaper**
- Consensus is **32.80%** more expensive



Task Name	Time per b.box	
	Median	Mean
Drawing	25.5s	50.8s
Quality Verification	9.0s	21.9s
Coverage Verification	7.8s	15.3s
Total	42.4s	88.0s

Evaluation – Quality Control

Drawing Task

- Acceptance ratio: **62.2%**

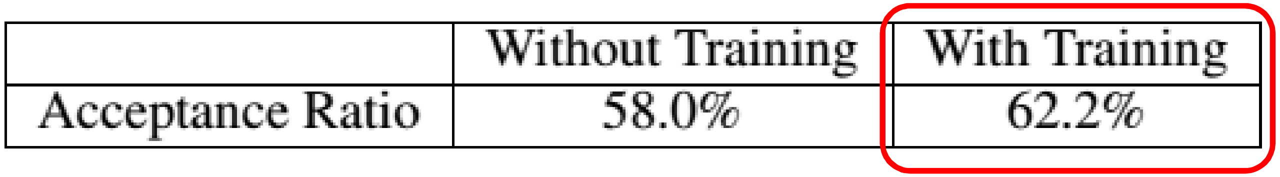
Quality Verification Task

- It was employed a “gold standard” (validation images)
- Acceptance ratio: **89.9%**

Coverage Verification Task

- It was employed a “gold standard” (validation images)
- Acceptance ratio: **95.0%**

Effectiveness of Worker Training



	Without Training	With Training
Acceptance Ratio	58.0%	62.2%

Conclusion

- It was presented a method that collects **bounding boxes** annotation using **Crowdsourcing**.
- It is composed by 3 tasks:
  - Drawing Task
  - Quality Verification Task
  - Coverage Verification Task
- It achieves **high quality** data with **low-cost**.

Questions

References

[1]. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 248-255). IEEE.

[2]. Su, H., Deng, J., & Fei-Fei, L. (2012, July). Crowdsourcing annotations for visual object detection. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*.