The Peekaboom Data Format

Luis von Ahn, Ruoran Liu and Manuel Blum
Computer Science Department, Carnegie Mellon University
{biglou, royliu, mblum}@cs.cmu.edu

ABSTRACT

Peekaboom is an entertaining web-based game that can help computers locate objects in images. People play the game because of its entertainment value, and as a side effect of them playing, we collect valuable image metadata, such as which pixels belong to which object in the image. The collected data could be applied towards constructing more accurate computer vision algorithms, which require massive amounts of training and testing data not currently available. In this note we describe how to make sense of the Peekaboom data.

INTRODUCTION

Most of the best approaches for computer vision rely on machine learning: train an algorithm to perform a visual task by showing it example images in which the task has already been performed. For example, training an algorithm for testing whether an image contains a dog would involve presenting it with multiple images of dogs, each annotated with the precise location of the dog in the image. After processing enough images, the algorithm learns to find dogs in arbitrary images. A major problem with this approach, however, is the lack of training data, which, obviously, must be prepared by hand. Databases for training computer vision algorithms currently have hundreds or at best a few thousand images — orders of magnitude less than what is required.

Peekaboom address the problem of constructing a massively large database for training computer vision algorithms. To construct such a database, we follow the approach taken by the ESP Game [1] and introduce a new game called Peekaboom. Peekaboom is an extremely enjoyable networked game in which, simply by playing, people help construct a database for training computer vision algorithms.

The goal of this note is to explain the data format of Peekaboom. Before going into the data format, however, we explain the game itself.

BASIC GAME PLAY

Peekaboom, as the name may suggest, is a game with two main components: "Peek" and "Boom." Two random players from the Web participate by taking different roles in the game — when one player is Peek, the other is Boom. Peek starts out with a blank screen, while Boom starts with an image and a word related to it.

The goal of the game is for Boom to reveal parts of the image to Peek, so that Peek can guess the associated word. Boom reveals circular areas of the image by clicking. A click reveals an area with a 20-pixel radius. Peek, on the other hand, can enter guesses of what Boom's word is. Boom can see Peek's guesses and can indicate whether they are hot or cold.

When Peek correctly guesses the word, the players get points and switch roles; play then proceeds on a new image-word pair. If the image-word pair is too difficult, the two players can "pass," or opt out, of the current image. Passing creates the same effect as a correct guess from Peek, except that the players get no points.

To maximize points, Boom has an incentive to reveal only the areas of the image necessary for Peek to guess the correct word. For example, if the image contains a car and a dog and the word associated to the image is "dog," then Boom will reveal only those parts of the image that contain the dog. Thus, given an image-word pair, data from the game yield the area of the image pertaining to the word.

Pings

Another component of the game are "pings" — ripples that appear on Peek's screen when Boom right-clicks on the image (see Figure 1). If two players were playing with the image on Figure 1, then many correct words are possible from Peek's point of view: elephant, trunk, tusk, ear. Suppose the correct word is "trunk." To get Peek to guess correctly, Boom can "ping" the trunk of the elephant by right-clicking on it. In doing so, Boom helps to disambiguate the trunk from the rest of the elephant.

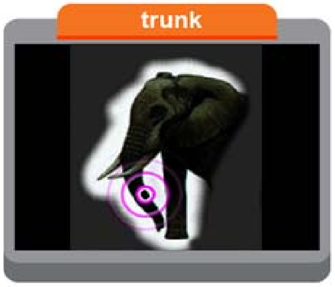

Figure 1. Pings. To help Peek, Boom can "ping" parts of the image by right-clicking on them.

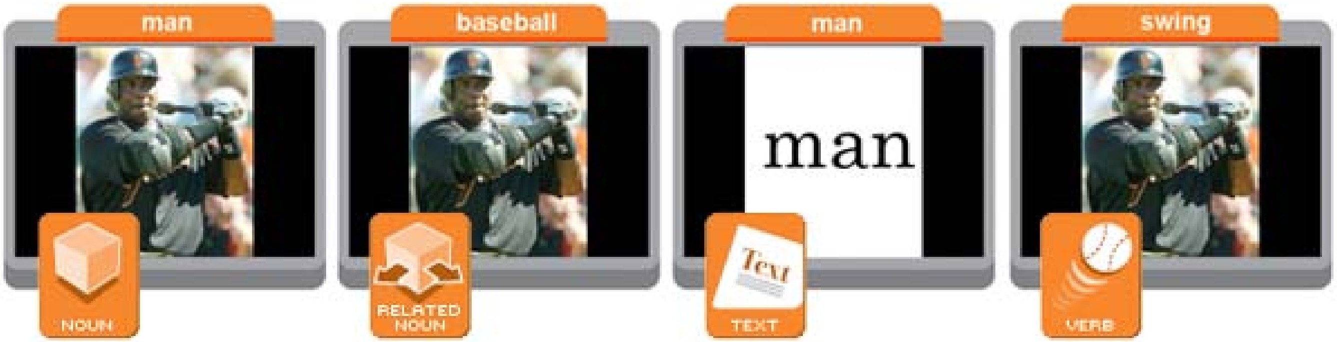

Figure 2. Hints. Boom can further help Peek by giving hints about how the word relates to the image: is it a noun describing something in the image, a noun related to the image, text on the image, or a verb?

Hints

Another feature of the game are buttons that allow Boom to give hints to Peek about how the word relates to the image (See Figure 2). Upon Boom's pressing of one of the hint buttons, a corresponding flashing placard appears on Peek's screen. The reason for having hints is that often the words can relate to the image in multiple ways: as nouns, verbs, text, or related nouns (something not in the image, but related to it).

The Bonus Round

Every time the players correctly complete four images, they are sent to a "bonus" round. The bonus round is different in nature from the rest of the game. In the bonus round, players simply click on an object in the image. The closer they are to each other's clicks, the more points they get. For example, both players could obtain an image of a car and be told "click on the car."

The Origin of Images and Labels

All words presented to the players are related to their corresponding image. On input an image-word pair, Peekaboom outputs a region of the image that is related to the word. We obtain millions of images with associated keyword labels from the ESP Game [1].

THE PEEKABOOM DATA FORMAT

Here we describe the Peekaboom Data Format.

First, there is a .pkb file for every image. Each line in a .pkb file is a specific play on an image, and everything is parenthesized heavily. Each line of the .pkb file has the following items in this order:

- (type of play). The type of play is either "n" (meaning normal round) or "b" (meaning bonus round).
- ((Image ID)(label)), gives the ID of the image being played on (which is the same as the filename of the image and the .pkb filename) and the label that was collected from the ESP Game.
- (timestamp), at the start of the round.
- (displayed dimensions of the image)
- (Boom_Player_ID Peek_Player_ID)
- (DATA), where DATA is a time-ordered sequence of events of the form: ((timestamp player) ((type of event) (data))). Here are some examples of events:
    - (8368 p)((x)()))) Means player p (for Peek) hit the pass button 8368 milliseconds after the image appeared on the screen.
    - ((1889 b)((b)(94 32))) means player b (for Boom) revealed a "blob" in coordinate 94,32 at time 1889 milliseconds.
    - ((5736 p)((g)(women))) means the guess "women" was entered by player p (for Peek) at time 5736.
    - ((1361 b)((h)(n))) means that player b gave a hint of type "n" (for Noun) 1361 milliseconds after the image appeared on the screen.
    - ((2245 b)((q)(0))) is a hot/cold event, which we will not explain in detail here.

It is important to note that the original dimensions of the image are *not* the same as the displayed dimensions of the image in the game. Therefore, before being used, the data needs to be rescaled back to the original image dimensions.

ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation (NSF) grants CCR-0122581 and CCR-0085982 (ALADDIN), and by a generous gift from Google, Inc. Luis von Ahn was partially supported by a Microsoft Research Graduate Fellowship.

REFERENCES

1. von Ahn, L., and Dabbish, L. Labeling Images with a Computer Game. In *ACM Conference on Human Factors in Computing Systems (CHI)*, 2004, pages 319-326.