Ask'nSeek: A New Game for Object Detection and Labeling

Axel Carlier[1], Oge Marques[2], and Vincent Charvillat[1]

[1] IRIT-ENSEEIHT, University of Toulouse, France
{Axel.Carlier,Vincent.Charvillat}@enseeiht.fr
[2] Florida Atlantic University, USA
omarques@fau.edu

Abstract. This paper proposes a novel approach to detect and label objects within images and describes a two-player web-based guessing game – Ask'nSeek – that supports these tasks in a fun and interactive way. Ask'nSeek asks users to guess the location of a hidden region within an image with the help of semantic and topological clues. The information collected from game logs is combined with results from content analysis algorithms and used to feed a machine learning algorithm that outputs the outline of the most relevant regions within the image and their names. Two noteworthy aspects of the proposed game are: (i) it solves two computer vision problems – object detection and labeling – in a single game; and (ii) it learns spatial relations within the image from game logs. The game has been evaluated through user studies, which confirmed that it was easy to understand, intuitive, and fun to play.

1 Introduction

There are many open problems in computer vision (e.g., object detection) for which state-of-the-art solutions still fall short of performing perfectly. The realization that many of those tasks are arduous for computers and yet relatively easy for humans has inspired many researchers to approach those problems from a 'human computation' viewpoint, using methods that include crowdsourcing ("a way of solving problem based on a large number of small contributions from a large number of different persons") and games – often called, more specifically, "games with a purpose (GWAPs)" [1].

In this paper we propose a novel approach to solving a subset of computer vision problems – namely *object detection and labeling*[1] – using games and describe Ask'nSeek, a two-player web-based guessing game targeted at the tasks of object detection and labeling. Ask'nSeek asks users to guess the location of a small rectangular region hidden within an image with the help of semantic and topological clues (e.g., "to the right of the bus"), by clicking on the image location which they believe corresponds to (one of the points of) the hidden region. Once enough games have been played using a given image, our novel machine learning algorithm combines user-provided input (coordinates of clicked points and spatial relationships between points and regions – 'above', 'below', 'left', 'right', 'on', 'partially on', or 'none') with results from off-the-shelf computer vision algorithms applied to the image, to produce the outline (bounding

[1] In this paper we use the phrase *object labeling* to refer to the process of assigning a textual label to an object's bounding box.

A. Fusiello et al. (Eds.): ECCV 2012 Ws/Demos, Part I, LNCS 7583, pp. 249–258, 2012.

box) of the most relevant regions within the image and their associated labels. These results can be compared against manually generated ground-truth (if such information is available) or used as semi-automatically generated ground truth for researchers in associated fields. Figure 1 shows examples of object detection and labeling results for two images from the PASCAL VOC 2007 dataset.

Fig. 1. Examples of object detection and labeling results obtained with the game-based approach described in this paper: (left) four objects /regions were detected and their bounding boxes were labeled as 'woman', 'sky', 'motorbikes', and 'man'; (right) two objects ('cat' and 'dog') were detected and labeled

2 Related Work

The idea of using games with the purpose of collecting useful data for computer vision has been brought first by Luis von Ahn and his ESP game [2]. In that game, two players are paired randomly and assigned the task of looking at the same image and typing keyword descriptions of the image. They score points when they manage to type the same keyword; in that case the word becomes part of the tags describing the image. This game has been initially devised to address the problem of constructing ground truth database for training computer vision algorithms. In the same spirit, Peekaboom [3], a subsequent and complementary game, goes a step further since it consists in locating objects (labeled by ESP) in a given image. Two players are again paired randomly: while one player reveals parts of the image, the other (who initially sees nothing from the image) has to guess the correct associated label.

In 2009, Ho et al. postulated that the cooperative nature of the ESP game has a number of limitations, including the generation of less specific or diverse labeling results, and proposed a competitive game for image annotation: KissKissBan [4]. Their game uses a *couple*, whose objective is the same as the players in the ESP Game (i.e., to guess what the partner is typing), but introduces the role of *blocker*, a third party who has 7 seconds to provide a list of blocked words, which contains the words he thinks couples might match on. They show that the results from their game have higher entropy than the ones produced by the ESP game (used as baseline for comparison), and are, therefore, more diverse.

More recently, Steggink and Snoek [5] presented the Name-It-Game, an interactive region-based image annotation game, whose labels are semantically enhanced by means

of the WordNet ontology. Name-It is a two-player game in which players switch roles (either *revealer* or *guesser*) after each turn. The revealer is shown an image and a list of words, from which he selects an object name, chooses the definition (obtained via WordNet) that best describes the sense in which that word is used in that particular image, and outlines the object of interest using a combination of polygonal and freehand segmentation, in order to progressively reveal an object in an image to the guesser. The guesser has to guess the name of the object (or a synonym) and may ask for hints during the guessing process.

In another recent effort, Ni et al. [6] have designed P-HOG (Purposive Hidden-Object-Game), a single-player game in which the goal is to locate an object that has been artificially embedded (i.e., hidden) within an image by drawing a bounding box around it.

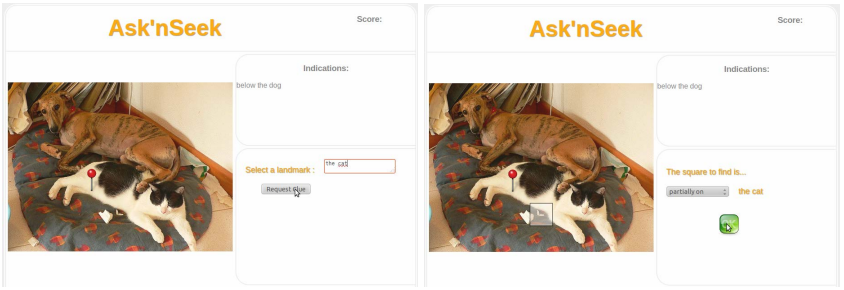
The main difference between Ask'nSeek and all of the above-mentioned games is that it does not require any player to explicitly outline regions or objects (or draw bounding boxes around them). Most importantly, Ask'nSeek is better than any of its predecessors in the sense that our game was designed to conceal the desired tasks expected to be performed by the users (labeling regions, clicking on relevant points within the image, and establishing meaningful spatial relationships between points and regions) while keeping it quick and entertaining.

3 The Game

3.1 Basic Structure, Terminology, and Rules

Ask'nSeek is a two-player, web-based, game that can be played on a contemporary browser without any need for plug-ins. One player, the *master* (Figure 2(b)) hides a rectangular region somewhere within a randomly chosen image. The second player (*seeker*) (Figure 2(a)) tries to guess the location of the hidden region through a series of successive guesses, expressed by clicking at some point in the image. What makes the game more interesting is that, rather than just blindly clicking around, the seeker must ask the master for clues relative to some meaningful object within the image before each and every click. Once the master receives a request for a clue from the seeker containing a label, it is *required* to provide a spatial relation, which is selected from a list: {above, below, to the right of, to the left of, on, partially on, none of the above}. These *indications* – in the form of (spatial relation, label), e.g., "below the dog" – accumulate throughout the game and are expected to be jointly taken into account by the seeker during game play. Based on the previously selected points and the indications provided by the master, the seeker can refine their next guesses and – hopefully – guess the hidden region after relatively few attempts. The game is played in *cooperative mode*, i.e., the master wants the seeker to locate the region as quickly as possible, which usually leads to accurate clues and game logs with high quality information.

According to the classification in [1], Ask'nSeek is an "Inversion-Problem game", because "given an input, Player 1 (in our case, called *master*) produces an output, and Player 2 (the *seeker*) guesses the input". More specifically, the *input* in question is the location of the hidden region within an image and the *outputs* produced by Player 1 are what we call *indications*.



(a) (b)

Fig. 2. Screenshots of the Ask'nSeek game: (a) seeker's screen; (b) master's screen

- **Initial Setup:** Two players are randomly chosen by the game itself.
- **Rules:** The master produces an input (by hiding a rectangular region within an image). Based on this input, the master produces outputs (spatial clues, i.e., indications) that are sent to the seeker. The outputs from the master should help the seeker produce the original input, i.e., locate the hidden box.
- **Winning Condition:** The seeker produces the input that was originally produced by the master, i.e., guesses the correct location by clicking on any pixel within the hidden bounding box.

3.2 Interpretation of Game Logs through a Machine Learning Algorithm

The machine learning strategy adopted in our work lies within the "semi-supervised clustering with constraints" framework. It combines data from two main sources: game logs and output of suitable computer vision algorithms. The game logs contain labels as well as 'on', 'partially on' and 'left-right-above-below' relations. Examples of labels include foreground objects (e.g., dog, bus) as well as other semantically meaningful regions within the image (e.g., sky, road).

We employ various content analysis algorithms (e.g., bottom-up saliency maps, interest point detectors) to derive a set of points that we will try to cluster in our model. For example, if we take as an input a saliency map, we randomly choose points following the distribution described by the saliency map. The goal of our algorithm is then to estimate a mixture of Gaussians that best describes our set of points, in which each resulting 2D Gaussian is assigned a label obtained from the game logs.

The indications given by players are used in different ways, depending on their type:

- the 'left-right-above-below' relations are used to create starting bounding boxes and can be seen as "hard constraints", i.e., the associated Gaussian can never be contradictory with these relations;
- the 'on' relations help us initialize the position for a Gaussian; and
- the 'partially on' relations can be seen as a "soft constraint" and provide information on the limits of a Gaussian. We use it to limit the size of the corresponding Gaussian, i.e. to constrain to the growth of the associated bounding boxes.

We grow the Gaussians and force them to respect the constraints described above. When the algorithm stops we compute a bounding box for each Gaussian and (for visualization purposes) overlay it on the image with its associated label.

4 Evaluation

In this section we describe several steps used to evaluate the feasibility of the approach, the minimum number of games needed to produce enough information for the underlying machine learning algorithm, and the quality of results obtained on images for which enough games have been played.

4.1 Simulating Game Logs – Experiments with Synthetic Data

After having conceived, designed, and implemented the Ask'nSeek game and performed a preliminary user study that showed that it is potentially fun to play, we proceeded to assess the quality of the data that can be collected and inferred from game logs. To do so, we decided to simulate a large number of game logs and analyze the generated traces, first by taking everything into account and then by limiting ourselves to a tiny fraction of the total number of simulated games.

Simulation Principles. We designed a game simulator whose goal was to enable us to quickly acquire a large amount of ready-to-use data (i.e., game logs) without having to deploy the game at a large scale and collect data from many users. Moreover, as a bonus, we might achieve a deeper understanding of how the game data enables our machine learning algorithm to do its job which, consequently, might lead to improvements and refinements of the game itself. The game simulator makes several important assumptions, among them: (i) the master never lies about the spatial relationship between the hidden region and a labeled region in the image; (ii) the seeker never makes mistakes, such as clicking on a pixel that should be ruled out due to previously received clues (this assumption also implies that the seeker has "perfect memory" and takes *all* previous clues into account before guessing the location for the next click); (iii) we have complete control over the labels, i.e., they come from a preselected vocabulary (consistent with the ground truth annotations for the PASCAL VOC dataset) and they do not contain any noise, misspellings, etc.; and (iv) neither master nor seeker "gives up" before having attempted all feasible options.

The first step of the simulation consists in the generation of a ground truth segmentation and labeling of the image. We have adopted the same conventions and terminology used in the PASCAL VOC dataset to associate each object to its surrounding bounding box and a label, and extended it to background elements such as sky, road, etc. This was based on observations from the first user study, where users reported that they tend to use *all the information present in the image* – rather than just the objects associated with PASCAL VOC object detectors – to find the hidden region more easily and quickly.

Once we have produced ground truth for an image, we generate simulated player traces for each simulated game, using an algorithm that models all the typical steps during game play, from the master's choice at the beginning, to the game logic used to

determine if there is a winner or not and whether there is any clue that the master may still provide to the seeker.

Assessing the Impact of the Number of Games. We generated 10,000 game simulations for every test image. Each entry in the game log associates the coordinates of a point, a spatial relation, and a label.

First we considered only the game logs that use the spatial relations "above", "below","on the left of" and "on the right of". We then used that information to build a bounding box limiting the region that respects all the constraints defined by these relations, within which the object must reside. Figure 3(a) plots in black all the points that fall outside this bounding box for the 'dog' object.

(a) (b)

Fig. 3. Analysis of simulation logs with different number of simulated games: (a) 10,000 games; (b) 7 games. See text for details

Second, we augmented the amount of information provided by the bounding box by incorporating the spatial relation "on". The information provided by these points is very strong, because they state with certainty that a point which is "on" an object is actually part of it. Rather than just using the (x, y) coordinates of the "on" points, we use the SLIC superpixel segmentation algorithm implementation from [7] to split the image into subregions, i.e., to "grow" each "on" point to its corresponding superpixel. By doing so, we consider not only the point, but the entire segmented region it belongs to, as being "on" the object. Such regions are plotted in red on Figure 3. We apply the same treatment to "partially on" points and plot the corresponding superpixels in green on Figure 3.

Figure 3(a) shows that the combination of superpixels corresponding to "on" and "partially on" points from 10,000 game logs produces an almost perfect rectangular bounding box. For the sake of comparison, Figure 3(b) shows the equivalent points if only a very small subset of our simulated game logs, in this case seven games, is taken into account.

4.2 Examples of Results from Actual Game Logs

After deploying the game and collecting actual game logs, we performed a preliminary (mostly qualitative) evaluation of the object detection and labeling results obtained using the proposed approach. Figure 4 shows the direct outputs of our model (where the

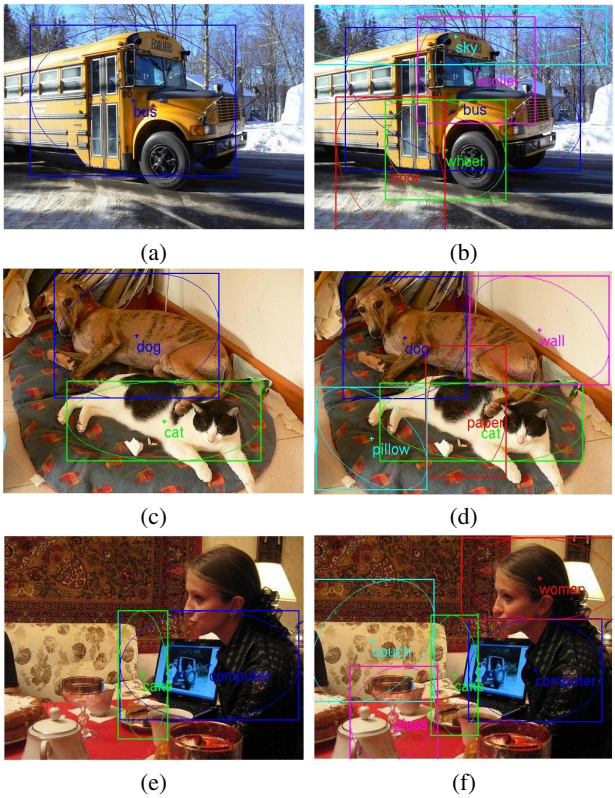


Fig. 4. Representative outputs of our model for three images of increasing visual complexity: (left column) dominant labels only; (right column) 5 most frequent labels. See text for details.

final bounding boxes enclose the Gaussian ellipses produced by the machine learning algorithm), once it has been applied to real traces for three different PASCAL VOC images: `2007_003137`, `2007_002597`, `2007_002914`. In a sense, these three images present an increasing visual richness: one bus, two pets and many objects. We collected 19 games and 56 indications for the bus image, 17 games and 44 indications for the cat and dog image and 19 games and 57 indications for the woman. As an example, the 44 indications for the cat and dog image are made of 2 'above' indications, 13 'below', 10 'left', 5 'right', 9 'on' and 5 'partially on'. The average length of the games is $44/17 = 2.6$ indications which actually shows that Ask'n'Seek finishes rather quickly on this image. The average number or indications per game is 3.0 for the two other images.

Figure 4 allows us to compare the results produced by our model in two distinct cases: using only the most cited labels (on the left column) and with the 5 most cited labels (on the right). For the bus image, the most cited label is *bus* (24 occurrences), followed by *wheel* (8), *door* (7), *sky* (5) and *"ecolier"* (5 occurrences). Figure 4(a)-(b) highlight the existing interactions between clusters: when five labels are used (b), the

size and shape of the dominant cluster (*bus*) changes a bit, when compared to the result for only one label (a). For the cat and dog image, the most cited labels, by far, are *dog* (used 16 times) and *cat* (16 times as well). By only handling these two dominant labels (the next most cited labels are cited 5 times or less), we obtain the result shown in Figure 4(c). In addition, we extracted richer labels from the game data, namely: 5 occurrences of *head* (2 *cat's head*, 3 *dog's head*), as well as 5 *legs* and 2 *nose* labels combined with *cat*, *dog*, *front, back* in a more complex way, which naturally corresponds to multiple instances (of heads, noses, etc.). In our current implementation, we don't handle these composite labels for 'parts of objects'. When the label *cat's head* is cited, it increases the count of *cat's head* occurrences as well as the count for *cat*, since this label is already dominant. As a consequence, the 3 next labels presented in Figure 4(d) are *paper* (4 occurrences), *wall*, and *pillow* (3 occurrences each). The results for label *wall* are reasonably good despite the fact that we didn't collect any 'on' points for it, i.e., the Gaussian cluster for *wall* simply fits within its bounding box. Figure 4(e)-(f) show the result for an image for which no obvious object emerges. Many labels are cited with almost the same frequency. We collected 8 occurrences of *computer*, 7 of *cake*, 6 of *woman*, 6 for *couch* and 5 for *glass* as well as fewer occurrences of *tea pot, cup of coffee, carpet, nose, head* etc. We chose to highlight the two most frequent ones (*computer* and *cake*) in part(e), and extend to include *woman*, *couch* and *glass* in part(f).

4.3 Comparison against a Baseline Object Detector

In this subsection we show a preliminary visual (i.e., qualitative) comparison between the results obtained with the Ask'nSeek game (with information from only 17 games) and the results produced by a state-of-the-art object detection algorithm, namely the "Discriminatively Trained Deformable Part Models" approach [8][2]. Figure 5 show representative results for the 'dog' and 'cat' objects and illustrate how our approach reduces the total number of false positives and improves the overall quality (i.e., size and location) of the bounding boxes.

5 User Studies

In this section we report the results of a preliminary evaluation after having enlisted 40 users to play the game, and highlight results obtained from these game logs.

Here is the protocol we followed for the user study.

1. The game is web-based and implemented in HTML5, i.e., no plug-in is required to play. We use a classical client/server architecture, in which the server handles the communications between the players (i.e., the clients) as well as the flow of the game and persists players' interactions into a database.

[2] We used the MATLAB code available at [9], which contains the official implementation of [8]. We followed the instructions provided by the authors and left the threshold parameter unspecified (default option).

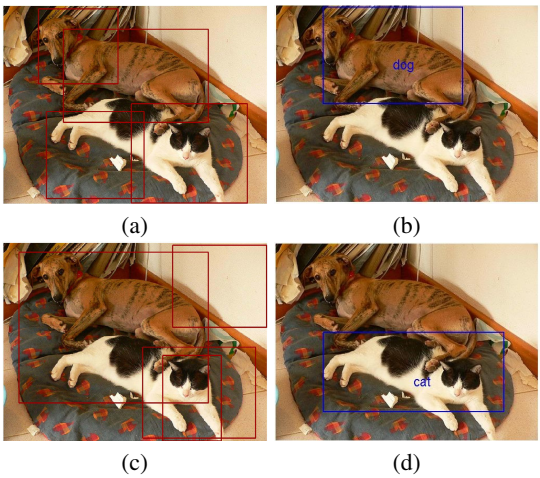


Fig. 5. Representative representative results: (a) baseline dog detector from [9]; (b) result from our approach for label 'dog'; (c) baseline cat detector from [9]; (d) result from our approach for label 'cat'

2. We used ten images from the PASCAL VOC dataset. This dataset was chosen because of its popularity for benchmarking in object detection and related tasks and for its public availability. After a sequence of games is played, we randomly permute the images.
3. We had a tutoring process, during which the game was explained to users (in a computer lab setting). Both master and seeker roles were described in detail, and game aspects such as the exact meaning of each spatial relation were carefully explained.
4. We collected data from 40 participants (25 males and 15 females), with ages ranging from 18 to 62. Each game requires a pair of participants. Each pair is allowed to play as many games as they desire. The total number of games played in this first user study was 148, with an average of 3.1 indications per game.
5. Players made use of all the spatial relations they were provided with. 'On the left' represents 18% of the indications, 'on the right' 19%, 'above' 15%, 'below' 19%, 'on' 13%, 'partially on' 12% and finally 'can't relate' 4%.
6. At the end of the process, users were asked to evaluate the game on four major aspects – enjoyability, simplicity, ergonomics and clarity – using a Likert scale, ranging from 'very good' (5) to 'very bad' (1). These are the results: enjoyability: 3.6; simplicity: 3.9; ergonomics: 3.4; and clarity: 3.7. In addition, users were asked if they would be interested in playing again at a later time, to which 40% answered "Yes", 45% said "Why not?", and 15% replied "No".

In summary, most of the players found the game enjoyable and fun to play.

6 Conclusions

This paper proposed a novel approach to solving a selected subset of computer vision problems using games and described Ask'nSeek, a novel, simple, fun, web-based guessing game based on images, their most relevant regions, and the spatial relationships among them. Two noteworthy aspects of the proposed game are: (i) it does in *one game* what ESP [2] and Peekaboom [3] do in *two* games (namely, collecting labels and locating the objects associated with those labels); and (ii) it avoids explicitly asking the user to map labels and regions thanks to our novel semi-supervised learning algorithm.

We also described how the information collected from *very few* game logs per image was used to feed a machine learning algorithm, which in turn produces the outline of the most relevant regions within the image and their labels.

Our game can also be extended and improved in several directions, among them: different game modes, timer(s), addition of a social component (e.g., play against your Facebook friends), extending the interface to allow touchscreen gestures for tablet-based play, and incorporation of incentives to the game, e.g., badges or coins, which should – among other things – encourage switching roles (master-seeker) periodically.

References

1. von Ahn, L., Dabbish, L.: Designing games with a purpose. Commun. ACM 51, 58–67 (2008)
2. von Ahn, L., Dabbish, L.: Esp: Labeling images with a computer game. In: AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors, pp. 91–98 (2005)
3. von Ahn, L., Liu, R., Blum, M.: Peekaboom: a game for locating objects in images. In: CHI, pp. 55–64 (2006)
4. Ho, C.J., Chang, T.H., Lee, J.C., Jen Hsu, J.Y., Chen, K.T.: Kisskissban: a competitive human computation game for image annotation. SIGKDD Expl. 12, 21–24 (2010)
5. Steggink, J., Snoek, C.G.M.: Adding semantics to image-region annotations with the name-it-game. Multimedia Systems 17, 367–378 (2011)
6. Ni, Y., Dong, J., Feng, J., Yan, S.: Purposive hidden-object-game: embedding human computation in popular game. In: ACM MM 2011, pp. 1121–1124 (2011)
7. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), `http://www.vlfeat.org/`
8. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. PAMI 32, 1627–1645 (2010)
9. Felzenszwalb, P., Girshick, R., McAllester, D.: Discriminatively trained deformable part models, release 4, `http://people.cs.uchicago.edu/~pff/latent-release4/`