Multimedia Systems
DOI 10.1007/s00530-010-0220-y

INTERACTIVE MULTIMEDIA COMPUTING

Adding semantics to image-region annotations with the Name-It-Game

Jeroen Steggink · Cees G. M. Snoek

Abstract In this paper we present the Name-It-Game, an interactive multimedia game fostering the swift creation of a large data set of region-based image annotations. Compared to existing annotation games, we consider an added semantic structure, by means of the WordNet ontology, the main innovation of the Name-It-Game. Using an ontology-powered game, instead of the more traditional annotation tools, potentially makes region-based image labeling more fun and accessible for every type of user. However, the current games often present the players with hard-to-guess objects. To prevent this from happening in the Name-It-Game, we successfully identify WordNet categories which filter out hard-to-guess objects. To verify the speed of the annotation process, we compare the online Name-It-Game with a desktop tool with similar features. Results show that the Name-It-Game outperforms this tool for semantic region-based image labeling. Lastly, we measure the accuracy of the produced segmentations and compare them with carefully created LabelMe segmentations. Judging from the quantitative and qualitative results, we believe the segmentations are competitive to those of LabelMe, especially when averaged over multiple games. By adding semantics to region-based image annotations, using the Name-It-Game, we have opened up an efficient means to provide precious labels in a playful manner.

Keywords Image-region annotation · Ontology · Labeling game

J. Steggink · C. G. M. Snoek (✉)
University of Amsterdam, Science Park 904,
1098 XH Amsterdam, The Netherlands
e-mail: cgmsnoek@uva.nl

1 Introduction

To unlock the ever increasing collections of digital image data, it is required to obtain semantic access to the image content. A semantic meaning can be allocated to an entire image, but an image is usually composed of various regions and every region can have a different semantic meaning. It is therefore useful for many applications to label the semantic meaning per region. Two types of labeling solutions have emerged: (1) the first approach is machine-driven with automatic assignment of labels to image regions; (2) the second approach to the labeling problem relies on human labor, where labels are assigned manually after visual inspection. Automatic image annotation methods aim to establish a relation between the low-level features, derived from the image region, and the semantic meaning the region would have for a human. See [1, 2] for an overview. To learn the relation between features and semantics, the current paradigm of choice is to rely heavily on supervised machine learning and annotated training examples. Despite constant progress in this area, automated region segmentation remains an unsolved problem. One of the main obstacles is obtaining the annotated training images. Since both automatic and human solutions for semantic access to image regions depend on manual labeling, this paper focusses on the manual image annotation process in more detail.

Manual labeling of images has traditionally been the realm of professionals. In cultural heritage institutions, for example, library experts label archival photos for future disclosure using controlled vocabularies [3]. Because expert labeling is tedious and costly, it typically results in a brief description of a complete image. In contrast to expert labor, Web 2.0 has launched social tagging, a recent trend to let amateur consumers label, mostly personal, visual

Published online: 09 December 2010

content on web sites like Picasa, Flickr, and Facebook. Since the labels were never meant to meet professional standards, amateur labels are known to be ambiguous, overly personalized, limited, and often error prone [4, 5]. Furthermore, the annotations are hardly ever defined on a region level.

To cater for region-labeled image collections, e.g., for the purpose of object detection, research initiatives like LabelMe [6] and many others have emerged recently [7–9]. In contrast to global image annotation collections, e.g., [10–12], region-based annotation collections only contain a limited number of classes. To extend the annotation vocabulary, Barnard et al. [7] presented a data set with 1,014 region-based annotations linked to the WordNet ontology [13], a lexical database in which nouns, verbs, adjectives, and adverbs are organized into synonym sets (synsets) based on their meanings and use in natural language. With the aid of ontologies, the objects in the images can be annotated in a more semantic-rich manner [14, 15]. Using an ontology, a machine might even be able to reason based on the connections between objects and properties within a knowledge domain. An example of the assistance that an ontology would be able to provide is when there is a car visible in the image. When, for example, the relation between a car and wheels has been established in an ontology, this relation could be used to recognize the car earlier [16]. By adding the semantic structure of an ontology to the manual annotation process, the region-based image labels become more descriptive and thus more useful.

Apart from the lack of semantic structure, we attribute the limited number of publicly available region-based image labels to current annotation tools. The level of complexity of most tools, and hence, the time necessary to perform the region-based image annotation, complicate the fast annotation of large numbers of objects. To create a large data set of annotated images quickly, with as many objects as possible, von Ahn [17–19] developed several games to replace the task of the conventional annotation tools. In his games, people easily allocate keywords to selected images and objects. People are stimulated to play the game by awarding them points when two players agree on a label, leading to a high score. von Ahn has shown that by using such an annotation game, images can be quickly annotated (region-based) without the players noticing what task they are really doing. However, in all these games no semantics are added to the annotations.

In this paper, we propose the Name-It-Game which ultimately aims to create a large data set of region-based semantically annotated images. By linking as many objects as possible to an ontology, this game will make it feasible to annotate region-based images both semantically and quickly. The research question we study is: which interactive multimedia system components should be present in an annotation tool to enable the quick labeling of different objects in images, in a semantically and region-based manner? To answer this question, we first analyze in depth related annotation tools in Sect. 2. Based on the analysis of these tools, including their advantages and disadvantages, we introduce the Name-It-Game in Sect. 3. We present the experimental setup in which we evaluate the core components of the Name-It-Game against existing annotation alternatives in Sect. 4. We discuss the experimental results in Sect. 5. Finally, we conclude in Sect. 6.

2 Related annotation tools

After von Ahn's introduction of the Extra Sensory Perception (ESP) game [17], many multimedia annotation games have been proposed. Examples include music [20, 21], video [22, 23], and 3D annotation games [24, 25]. In our review of related work, we restrict ourselves to labeling games and other annotation tools for 2D images. We review existing work on image annotation tools by structuring our discussion according to four principal questions related to the annotation process: How to annotate?, What to annotate?, Why annotate?, and Who annotates? Starting from these questions, the most important components of the existing image annotation tools are described and analyzed.

2.1 How to annotate?

In the literature we distinguish three different types of tools for annotation. The conventional annotation tools where the tool runs on the desktop, the online tool that can be accessed via the Internet with a web browser, and the online games. The advantage of the annotation tools that run on a desktop PC [26, 27, 29] is that they can support various components that take a great deal of processor power, such as basic image processing functionality [27, 29]. The disadvantage is that the program has to be installed and is often platform dependent. The advantage of online tools [6, 30, 31] is that they are easily accessible and that they have a larger public at their disposal to provide the annotations. However, a disadvantage is that the processor-intensive components cannot easily be incorporated. Online games [17–19] have the potential to reach a far larger public than the desktop and online annotation tools. Annotation takes place quickly, but the disadvantage is that the quality is often inferior to conventional annotations. Hence, it is important to develop an image annotation game in such a manner that the game elements result in increasing the quality of the annotations to an as high a level as possible.

The faster the annotations can be executed, the faster the data set will grow. With the desktop and online tools it often takes a considerable amount of time to perform an annotation. With the M-OntoMat-Annotizer tool [27], for example, the object must first be selected in the ontology, then the image must be opened, and next the object in the image must be outlined. It will take a user a few minutes to perform these actions, but when done properly, it yields high-quality annotations. When this process is quickened, it will decrease in annotation quality, but it will make it possible to do a large number of annotations within less time. The online games are a good example of this process. One annotation typically takes less than a minute. Besides the games, only the IBM EVA [31] tool offers fast annotations. One of the reasons is that this tool does not offer the possibility of object-segmentation in its annotation process.

In order to add semantics to annotations, ontologies such as WordNet [13] may be used. Six of the ten annotation tools we have evaluated use an ontology. This component is present in all desktop versions [26–29], the online tool IBM EVA [31], and LabelMe [6]. In general, there is an ontology browser present in the desktop tools, which makes it possible to select the correct class or instance for the object of interest. When the same objects are repeatedly annotated, it will take only a short time for a user to find the object in the ontology. When another class or instance has to be found repeatedly, it can sometimes take a long time before the user finds the correct instance in the ontology. In LabelMe the annotations are linked to WordNet [6, 13]. An effort is made to find the correct sense in WordNet automatically and to connect this sense to the annotation. However, many words have more than one meaning (sense), which makes it difficult to choose the correct word with the correct meaning. For this reason, the data set administrators choose the sense manually at the end of the annotation process. Unfortunately, this function has not been incorporated in the online annotation tool, nor is the WordNet ‘synsetid’ present in the data set.

When annotations are too abstract, the object can often not be assigned to an image-region. This means that it is hard to allocate a word describing activities, feelings, emotions etc. to an object. For the desktop tools this is not so much of a problem, since the annotators are experts and choose their words carefully. On the other hand, in the games this is problematic. The images that are annotated in ESP Game [17] are used as input for Peekaboom [18] and Squigl [19]. However, the annotations that are provided by ESP Game are not filtered on abstraction. Examples of abstract words are: love, tennis, joy and nice. Although these words do have added value as an annotation, the specific location of these words cannot, or only with difficulty, be identified in the image. To prevent this from happening, ideally, these word categories must be filtered out.

2.2 What to annotate?

Nearly all image annotation tools offer the possibility to make region-based annotations. The only exceptions are IBM EVA [31] and the ESP Game [17]. There are various methods to execute region-based annotations: bounding box [17, 27, 28, 30, 31], polygonal [6, 26] and freehand [18, 19, 27] drawing. With the bounding-box technique the object is framed by dragging a rectangle around an object. The advantage of the bounding-box selection is that it is very fast; however, the disadvantage is that the selection is inaccurate and often selects much more image data than necessary. The polygonal method offers the possibility to make a more detailed selection by drawing a polygon around the object. This method is fast, and more precise than the bounding box. Nevertheless, since it uses straight lines, it is still difficult to make a very accurate selection. With a freehand drawing tool, one can draw a free line around an object, which enables very precise selections. The obvious disadvantage is that it takes longer to draw an accurate line around an object. Two of the annotation tools, Spatial Annotation Tool [29] and M-OntoMat-Annotizer [27], offer the option to use an automatic segmentation method. However, the quality of the automatic segmentations does not meet the high standard required for image-region annotations. Consequently, this component has not been included in the analysis. The two methods which offer both quick and detailed image-region segmentations are the polygonal and freehand drawing methods.

2.3 Why annotate?

Motivations for image annotation are many. In-depth studies on the topic, emphasizing in particular organizational and social aspects, are presented in [32, 33]. For annotation games, we also consider the scoring mechanism. This leads to three different reasons why the annotators annotate the images. These are: organizational [6, 26, 27, 31], social [6, 28–31], and scoring [17, 18, 19].

Generally, desktop tools are used by research groups. The purpose of the annotating is to organize the annotations for research purposes. Researchers have only limited time to annotate images. They often rely on students or an external bureau, such as Mechanical Turk [34] to provide the annotations. When researchers rely on students, the stimulus is either an interest in the research project or money. The online tools reach a larger group of annotators since the tools can be accessed by everybody via the Web. LabelMe [6], for example, appeals to a large group of annotators. The idea is that fellow-researchers in the object-detection field help each other to create an ever larger data set for research purposes. However, at the moment it focuses too much on the unilateral purpose of annotation, and non-researchers are unlikely to carry out large numbers of annotations. Furthermore, the annotation speed is not quick enough. Social web sites such as Flickr [30] have been set up with the aim of sharing images with other people. The tags that have been included with the images facilitate image searches and present a potentially better representation of what the image shows, but tags are known to be ambiguous, overly personalized, and limited per item [4, 5]. Using games as annotation tools offers a whole new stimulus. By earning points and a high score, annotating suddenly has a different purpose to the user. This yields more annotators and is cheaper than hiring annotators. Of course, the number of annotations generated by using games is totally dependent on the popularity of the game. When an annotation game is unpopular it can be considered similar to an online annotation tool.

2.4 Who annotates?

We distinguish between two different groups of annotators: experts and amateurs. Theoretically, everybody is able to make annotations; however, in the literature we notice that, in general, only experts use the desktop tools. These programs require a thorough knowledge of the tool and how to use it for the annotation process. On the other hand, the online tools and games, with the exception of IBM EVA [31], make annotation accessible for every type of user. With the IBM EVA tool [31] it is relatively easy to provide annotations and it can be done fast; however, you have to be an expert to work with the tool. Both LabelMe [6] and the games are easy to use. With the games, annotations are made while playing, without the players really noticing what they are doing. The games make it possible to quickly provide a large number of annotations and are accessible to a large audience. The easier and more accessible an annotation tool is, the larger the chance that more annotations will be made. Finally, the more people have access to an annotation tool, the faster the annotated data set may grow.

2.5 Contribution of this paper

Our analysis of related work shows that present-day annotation tools have a number of flaws with respect to quick semantic image-region annotation, which we summarize in Table 1. We build on this related work, but to resolve the identified flaws we present in this paper the Name-It-Game. We discuss the contributions of the Name-It-Game along the lines of *How*?, *What*?, *Why*? and *Who*?

- *How?* In order to have a large number of people provide annotations a game will be used. Up to the present day, none of the games mentioned in Table 1 uses an ontology to add semantics to the annotations. Name-It-Game would be the first labeling game to make it feasible to link the region-based object annotations to an ontology. Using this ontology, the Name-It-Game makes an effort to filter abstract words in order to make as many presented objects as possible fit for annotation.
- *What?* Name-It-Game must be able to generate image-region annotations quickly with a reasonably high segmentation quality. A combination of freehand and polygonal segmentations offers the fastest and most

Table 1 Overview of popular annotation tools

Annotation tool	How?		What?				Why?			Who?	
	Type	Ontology	Global	Bounding box	Polygonal	Freehand	Organizational	Social	Scoring	Expert	Amateur
Image Parsing [26]	Desktop tool	✓	–	–	✓	–	✓	–	–	✓	–
M-OntoMat-Annotizer [27]	Desktop tool	✓	–	✓	–	✓	✓	–	–	✓	–
Photostuff [28]	Desktop tool	✓	–	✓	–	–	–	✓	–	✓	–
Spatial Annotation [29]	Desktop tool	✓	✓	–	–	–	✓	–	–	✓	–
Flickr [30]	Online tool	–	✓	✓	–	–	–	✓	–	✓	✓
IBM EVA [31]	Online tool	✓	✓	–	–	–	✓	✓	–	✓	–
LabelMe [6]	Online tool	✓	–	–	✓	–	✓	✓	–	✓	✓
ESP game [17]	Online game	–	–	✓	–	–	–	–	✓	✓	✓
Peekaboom [18]	Online game	–	–	–	–	✓	–	–	✓	✓	✓
Squigl [19]	Online game	–	–	–	–	✓	–	–	✓	✓	✓
Name-It-Game	Online game	✓	**–**	**–**	✓	✓	✓	✓	✓	✓	✓

We structure their most characterizing components according to the four main questions related to the annotations process: *How* to annotate?, *What* to annotate?, *Why* annotate?, and *Who* annotates? Note the lacking components with respect to quick semantic image-region annotation in all of the tools. To address these shortcomings simultaneously, we propose in this paper the Name-It-Game (denoted in bold) optimal manner to make the segmentations manually. The advantage of freehand is that it enables outlining detailed objects and the polygonal method offers the possibility to make straight lines quickly.

- *Why?* The advantage of a game is that the aim, the annotation, is not immediately clear to the players [17]. People play the game to score points and to win from other players. Visualizing the score of both players on the screen and having a high score leads to competitive behavior in the players. The data generated by people playing the game can be used for research purposes and shared with the entire world.
- *Who?* The desktop and online tools are usually linked to expert users. Like other games, Name-It-Game should appeal to a large and diverse public. Providing annotations should be easy for everybody.

We consider the inclusion of an ontology into the complete game-play of an image-region annotation game the main technical contribution of this work. We focus in particular on the multimedia system aspects and leave the human-centered factors for future work. In the next section, we will present the Name-It-Game and we will discuss in detail how we will resolve the flaws identified in related work.

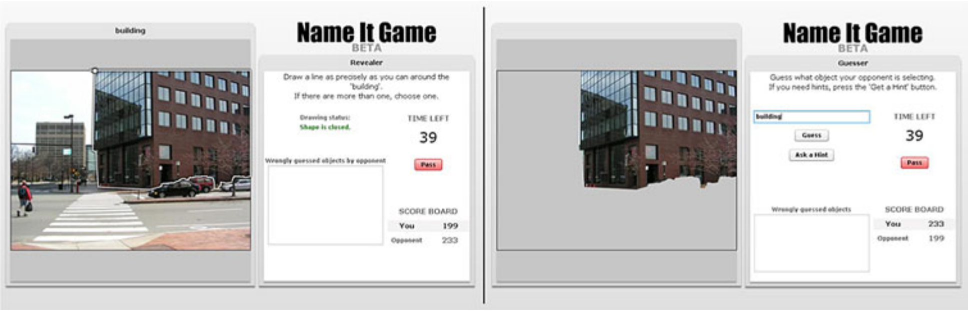
3 Name-It-Game

Name-It-Game is an image annotation game for two players. When players log in, they are linked to the first available competitor. Both players have their own separate role in the game which is switched after each turn: a player is either a *revealer* or a *guesser*. As *revealer*, the purpose of the game is to reveal an object in an image to the *guesser* who in turn has to guess which object it concerns, see Fig. 1.

3.1 How to annotate?

The online game functions as follows. The *revealer* is shown an image and a list of words, which we may obtain from various sources, such as photo sharing web sites, personal collections, or global image labeling games. Using global image labels as a source for a region-based labeling game introduces two problems. First, there is only a limited number of keywords available for the images. The objects as a whole are mentioned, but not their individual components. Second, the words are often too abstract for the purpose of outlining. We deem it (nearly) impossible to locate verbs, adjectives and adverbs as independent objects in an image. When used in combination with nouns, it is possible; however, since the keywords accompanying the obtained images often consist of a few words only, we consider it necessary to omit abstract words. To tackle these two problems, we enrich the Name-It-Game with semantic descriptions and semantic structure obtained from an ontology. As our ontology we use WordNet [13]. The ontology allows us to automatically supplement the list of objects shown to the *revealer* with object components, by relying on WordNet's "part-of" relationships. In this manner it becomes possible to obtain extra location information regarding the object components. In order to prevent that labels are too abstract, we also filter out the conceptual words using WordNet. We require that every object-label that is played in the Name-It-Game belongs to a WordNet category which consists of non-abstract nouns only. This results in the 12 word categories in Table 2. From this ontology-enriched list, the *revealer* must choose an object by clicking on the corresponding word, see Fig. 2a.

One of the requirements of the Name-It-Game is to obtain semantic annotations. It must be possible to establish a connection between the word and the object in the image. Since words often have more than one meaning, it is hard for a machine to know exactly which object it concerns. For example, take the word 'mouse'. It may just as well be an animal or a computer mouse. In order to establish which definition of the word it concerns, we ask the *revealer* to choose the correct definition of the chosen object from a list, see Fig. 2b. These definitions are again taken from WordNet. When the *revealer* has chosen the definition, the senseID of WordNet is saved in the game



Draw a line as precisely as you can around the 'building'. If there are more than one, choose one.

Drawing status: Shape is closed.

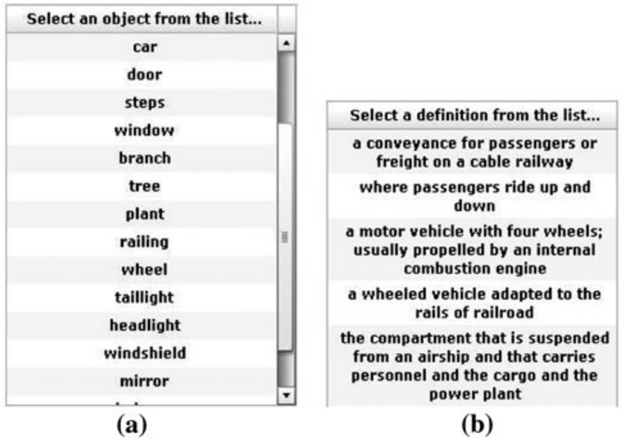
Guess what object your opponent is selecting. If you need hints, press the 'Get a Hint' button.

Fig. 1 User interface of the Name-It-Game. The player on the left is the *revealer* and outlines a selected object. The player on the right, the *guesser*, is shown the outlined object and must guess what it is

Table 2 List of WordNet category definitions used in the Name-It-Game

Category	Description
Noun.Tops	Unique beginner for nouns
Noun.animal	Nouns denoting animals
Noun.artifact	Nouns denoting man-made objects
Noun.body	Nouns denoting body parts
Noun.communication	Nouns denoting communicative processes and contents
Noun.food	Nouns denoting foods and drinks
Noun.group	Nouns denoting groupings of people or objects
Noun.object	Nouns denoting natural objects (not man-made)
Noun.person	Nouns denoting people
Noun.plant	Nouns denoting plants
Noun.shape	Nouns denoting two and three dimensional shapes
Noun.substance	Nouns denoting substances

This list prevents that the *revealer* selects a label which is too abstract for image annotation


Select an object from the list...

car
door
steps
window
branch
tree
plant
railing
wheel
taillight
headlight
windshield
mirror

Select a definition from the list...

a conveyance for passengers or freight on a cable railway
where passengers ride up and down
a motor vehicle with four wheels; usually propelled by an internal combustion engine
a wheeled vehicle adapted to the rails of railroad
the compartment that is suspended from an airship and that carries personnel and the cargo and the power plant

Fig. 2 Interactive mechanism to establish which definition of a word is played in the Name-It-Game. **a** List of objects from which the *revealer* must choose one, to segment in the image. **b** List of definitions from which the *revealer* must choose the correct definition for the chosen object. In this example the object 'car' is chosen from the list in **a**

database. Consequently, we know exactly which meaning of the word it concerns and with the aid of the structure of WordNet a machine can come to a conclusion regarding the object in an image. In order to safeguard players choosing the definitions blindly, we randomize the order of the definitions when presented to the *revealer*. In effect, forcing the *revealer* to look closely at the definitions before making a deliberate choice.

3.2 What to annotate?

When the *revealer* has chosen an object and the accompanying definition, the object needs to be outlined by drawing a line around the object in the image using the mouse, see Fig. 3a. By outlining the object it becomes slowly visible on the screen of the competitor, the *guesser*, see Fig. 3b. The first time that the player clicks on the image, a white dot appears, which is the starting point of the outline. To finish the outline, the *revealer* has to continue drawing the line until she arrives back at the starting point, the white dot. In the screen of the *revealer* it is also visible whether the outlining has been completed. This is indicated by status messages like 'Shape is not closed' and 'Shape is closed'. The Name-It-Game offers two ways to outline an object, straight lines where the objects are polygonal-like and a freehand line where high-precision is required. In the first method, the *revealer* segments the object in the image by repeatedly clicking on a point and so draw straight lines between the last two positions clicked on. In the second method, the *revealer* draws a line by continuously pressing down on the left mouse button at the place where the cursor is located. In this manner the player very accurately selects the objects. Naturally, it is also possible to use a combination of the two segmentation methods.

Since the *revealer* has no direct interest in providing useful high-quality image outlines, it is important to reward the segmentation quality by means of a scoring mechanism. Scores should be given for the ratio between line length and area. The smaller the area, the more points it should yield. This can be carried out logarithmical, resulting in a smaller increase in score with very small selections. So as to avoid that the *revealer* creates very small outlines, the player gains points for the speed with which the other player guesses the object. When the outline is very small, it will be very difficult for the *guesser* to guess the object. The *guesser* too should receive extra points for smaller objects, since they are harder to guess.

3.3 Why annotate?

Next it is the task of the *guesser* to guess which object has been displayed on his screen. He does this by entering the word of the object in a text box. Since the exact meaning of the word is already known when the *revealer* selects the word and the definition using the ontology, the *guesser* can mention the exact word, or one of its synonyms. This usage of the ontology avoids player frustration when the *guesser* cannot recall the appropriate synonym for the object. Moreover, it speeds up the game. The faster the object is guessed by the *guesser*, the more points should be awarded.

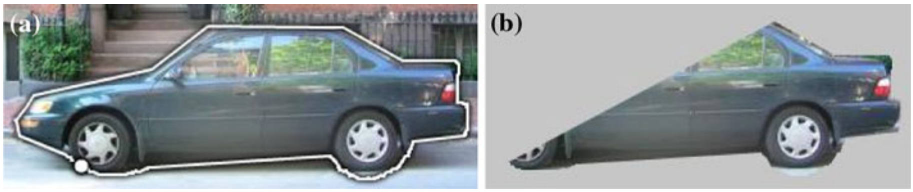
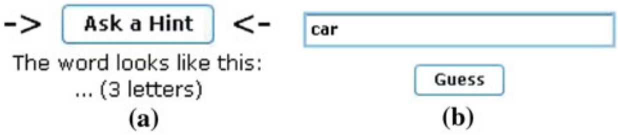


Fig. 3 Image segmentation in the Name-It-Game. **a** The *revealer* outlines the object 'car' using a combination of polygonal and freehand segmentation. **b** Gradual appearance of the object on the screen of the *guesser*



The word looks like this:
... (3 letters)

Fig. 4 Ontology-powered hint mechanism in the Name-It-Game. **a** The *guesser* asks for a hint and **b** labels the object as a 'car'

The *revealer* should earn points by outlining the object in as much detail as possible.

When the *guesser* experiences difficulty while guessing the object, he can ask for a hint by clicking on the button 'Ask a Hint', see Fig. 4a. The first hint the player receives is the number of letters of the word/object. Next, hints are given by exploiting, again, the semantic structure of WordNet by using the antonym, hypernym, hyponym, meronym, and the definition of the word. When there is no antonym, the hypernym is given, etc. The hints are also given in this order: from giving the least information about the object, to most information without literally telling the *guesser* the name of the object. To avoid turning the game into just a word game, where the *guesser* continues asking hints and guesses the answer without even looking at the object, asking hints should cost the *guesser* precious points. This does not mean that the hints cannot be used tactically. The faster the object is guessed, the more points the *guesser* receives. So ideally, one or two hints should help to guess the answer even faster and ultimately earn the *guesser* more points.

In order to provide the *guesser* with the correct hints, it is important that the *revealer* selects the correct definition. When the *revealer* selects an inappropriate definition, and the *guesser* asks for a hint, it is obviously difficult to link the hints to the object in the image, and consequently the chance that the object is guessed decreases. When the score of the *revealer* also increases when the *guesser* quickly guesses the answer, it is not in the interest of the *revealer* to choose an incorrect definition. When the *guesser* receives the correct hints, the object can be guessed faster. When asking hints costs points, this will be to the advantage of the *revealer*, but these points should be subtracted only after the word has been guessed. When the word is correct, the *guesser* should be given points for the time it has taken him to guess the word: the faster the guess, the more points rewarded. The roles of the players are reversed as soon as the object has been guessed by the *guesser*, see Fig. 4b.

3.4 Who annotates?

The Name-It-Game should be accessible and easy to understand for every type of player, experts as well as amateurs. By keeping the game simple and since it is intended as an online game accessible via the internet, it may well appeal to a wide audience.

4 Experimental setup

4.1 Game statistics

For the experiments about 80 people played the game using an online prototype, which used images from LabelMe [6]. Of these 80 people, 45 were registered users. The other 35 people were anonymous, so we do not know whether they are unique people. The registered players are from 17 to 60 years old. In total 1,558 games were played, of which 1,431 (92%) objects were guessed and 37 (2,4%) of the games were passed. The average number of word guesses is 1.48. Of the chosen definitions, only 64 (4.1%) were wrong. In 293 games (19%) hints were asked, with an average of 2.8 hints per game. This resulted in 231 (79%) guessed versus 62 (21%) not guessed objects. The average area of the segmentations where hints were asked is 9,137. The average area of segmentations of guessed objects is 12,617. In 135 (8.7%) games a synonym of the original word was guessed.

4.2 Experiments

In Sect. 3, we have identified the interactive multimedia system components needed for swift and semantic region-based image labeling using a game. In order to measure the effectiveness and efficiency of these components, we execute the following three experiments:

- *Experiment 1: adding semantics*
  To quantify the added value of adding semantics to region-based image annotations, we perform an

experiment on the labeling results of Peekaboom. In Sect. 2.1 we have explained that Peekaboom [18] and Squigl [19] offer the players objects which are hard to guess or to select. To find out what type of words are hard to guess or to select in the image, we analyzed 200 randomly chosen labeled objects from Peekaboom game data[1]. For each object we manually determined the WordNet category the word belongs to and the percentage of players that guessed the object. WordNet [13] has its words categorized in 45 categories. These are divided in 4 groups: verbs, adverbs, adjectives and nouns.

- *Experiment 2: labeling efficiency*
  To evaluate the efficiency of the Name-It-Game, we compare in experiment 2 the time needed to annotate an image. We compare the Name-It-Game with the only publicly available desktop tool that uses an ontology and has freehand selection support, namely: M-Ontomat-Annotizer [27]. Since M-Ontomat-Annotizer provides no support for WordNet (RDF format), we will use a sample ontology and measure the time needed to select an object in M-Ontomat-Annotizer and compare it with the time needed to select an object with the Name-It-Game. As we were not able to measure the time it takes to identify an object in WordNet using M-Ontomat-Annotizer, we also did not measure this (fast) process for the Name-It-Game. In effect, favoring M-Ontomat-Annotizer in this experiment. For the Name-It-Game, the timer started as soon as the *revealer* started drawing a line around the object and ended when the word was guessed. To carry out this experiment, 10 randomly selected images from the LabelMe set are used. The game will be played by 20 people. To test M-Ontomat-Annotizer, 1 of the 20 players who has worked with M-Ontomat-Annotizer before, will annotate the same 10 images. The time taken for opening each of the images and selecting the object regions will be measured.
- *Experiment 3: segmentation accuracy*
  To test the accuracy of the segmentations made with the Name-It-Game, we measure how close we can approach a 'gold standard' provided by the carefully verified LabelMe segmentations [6]. To be precise, we use a subset of 600 images and associated annotations from the LabelMe data set. All images and annotations we use are verified by administrators of the LabelMe team, but we had to filter some more keywords in the LabelMe annotations, i.e. keywords like 'carFrontal', were shortened to 'car' and (stemmed) keywords not found in WordNet were skipped. The LabelMe segmentations are accurate, and made using a polygonal selection tool. To compare the segmentations we will employ the same formula as von Ahn et al. used for Peekaboom [18]:

$$OVERLAP(A,B) = AREA\left(A \bigcap B\right) / AREA\left(A \bigcup B\right) \qquad (1)$$

We calculate the overlap ratio of a segmentation made with the Name-It-Game and a segmentation made with LabelMe. Where A is a Name-It-Game segmentation and B is a LabelMe segmentation. This experiment is divided in two parts. First we randomly select 50 unique played images with the Name-It-Game. We calculate the overlap ratio of these segmentations and the ones from LabelMe. This is to measure the accuracy after one play. Since we would have no ground truth for the annotations when the game is played, we are not able to pick just one segmentation and simply hope it is the best one. In the second part of this experiment we therefore test what the quality of a segmentation of the same object in the same image would be after a maximum of 10 plays. We have selected 10 different images at random and have 20 players play the Name-It-Game. In total each of the 10 images were segmented 10 times. To measure the accuracy of the segmentations after 10 plays, we measure how many times a pixel was selected. We determine if the combination of multiple selections leads to a high enough accuracy and how many pixels have to be selected to achieve this. We calculate the single best segmentation that comes closest to the LabelMe segmentation, the worst segmentation, and the best combined segmentation at what pixel count.

5 Results

5.1 Experiment 1: adding semantics

We summarize the results of adding semantics to region-based image labeling games in Fig. 5. The WordNet category of the top 11 of the most guessed objects in Peekaboom [18] corresponds with 10 of the 12 categories we suggested in Table 2. They all have a percentage guessed of at least 61.36% and a maximum of 80.70%. The noun.communication category ended up at the 13th place, with 40.00% guessed. The only missing category is noun.food, which was not amongst the 200 objects we chose randomly. One category, adj.pert (relational adjectives) also ended up in the top 11 with a guessed score of 66.67%. However, adjectives only give extra information to a segmentation if the noun is also known. Peekaboom presents the user with hard-to-guess word categories or hard-to-select objects in the image. These are the bottom 11 categories, plus adj.pert and verb.social in Table 2. These word categories should be avoided to improve the chance for labeling success. This experiment shows that the

[1] http://hunch.net/?p=170.

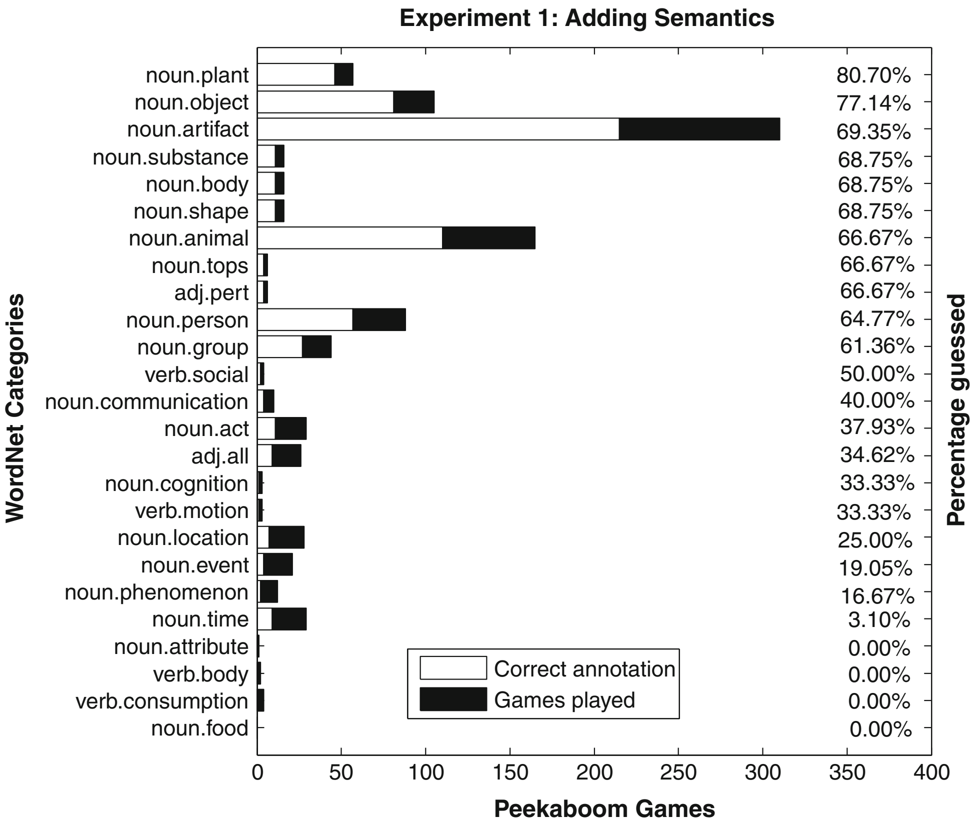


Fig. 5 Results for experiment 1: adding semantics to region-based image annotations. We group labels obtained from Peekaboom [18] in the appropriate WordNet category. The results show that words from the 12 WordNet categories we identified in Table 2 allow for the most guessable words. Hence, adding semantics to an image annotation game yields more useful labels

categories we carefully chose by analyzing labeling words, are indeed the categories which contain the most guessable words. Implying that embedding an ontology in an image labeling game yields more useful annotations.

5.2 Experiment 2: labeling efficiency

We summarize the results of experiment 2 for the labeling efficiency of the Name-It-Game in Table 3. Using the Name-It-Game, the average time taken to select a word and have it guessed by the *guesser* is 15.9 s, with a standard deviation of 11.5 s. The average time taken to open an image and select an object using the M-Ontomat-Annotizer is 23.7 s, with a standard deviation of 4.6 s. When measured using a Wilcoxon signed-rank test at the 0.01 level, the Name-It-Game is significantly more efficient than the M-Ontomat-Annotizer. Recall from Sect. 2.1, that in the M-Ontomat-Annotizer objects must first be selected in the ontology, then the image must be opened, and next the object in the image must be outlined, resulting in a sub-optimal labeling efficiency. The Name-It-Game is faster than M-Ontomat-Annotizer when selecting an object in the image, on average. In addition, for 8 out of 10 images the Name-It-Game results in more efficient image-region annotation. Only for small image regions, e.g., ‘pot’ and ‘speaker’, the M-Ontomat-Annotizer seems competitive with the Name-It-Game. It is of interest to note that, for this experiment, the segmentations made with M-Ontomat-Annotizer were of the same quality as with the Name-It-Game. In reality, M-Ontomat-Annotizer would be used by an annotation expert, who would make more precise segmentations at the expense of an even longer labeling time.

5.3 Experiment 3: segmentation accuracy

In the first part of the experiment, we have inspected the results of 50 randomly chosen games. An overview of the distribution of overlap ratios is summarized in the histogram in Fig. 6. Of the 50 randomly selected games, the average overlap ratio of the segmentations is 0.714, with a standard deviation of 0.180. The minimal overlap ratio was 0.199 and the maximum overlap ratio 0.930. This shows that overall the segmentations after one game are reasonably close to the LabelMe segmentations. We observe that the segmentations made with the Name-It-Game, in almost every case, are bigger than the LabelMe segmentations. We attribute this to the smaller images shown in the game, compared to the 3–5 times bigger images used in LabelMe. Moreover, in the Name-It-Game, segmentations are made quicker than segmentations made by an expert with a desktop or online annotation tool, which may result in less accurate segmentations.

Table 3 Results for experiment 2, labeling efficiency and experiment 3, segmentation accuracy

Image region		Labeling efficiency (s)		Segmentation accuracy (O)											
ID	Annotation	Baseline [27]	Name-It-Game	Best-case	Worst-case	1	2	3	4	5	6	7	8	9	10
(a)	Desk	30.1	**16.6**	0.93	0.29	0.82	0.90	0.93	**0.93**	0.93	0.93	0.92	0.92	0.92	0.84
(b)	Keyboard	25.3	**12.6**	0.94	0.82	0.58	0.71	0.78	0.83	0.87	0.90	0.91	**0.91**	0.89	0.19
(c)	Car	26.0	**14.5**	0.91	0.20	0.50	0.60	0.64	0.68	0.71	0.73	0.76	0.81	0.85	**0.91**
(d)	Apple	18.0	**11.5**	0.94	0.75	0.71	0.82	0.87	0.89	0.91	0.93	**0.93**	0.92	0.91	0.84
(e)	Mug	27.0	**21.7**	0.92	0.76	0.68	0.79	0.80	**0.80**	0.77	0.71	0.64	0.45	0.31	0.12
(f)	Sky	28.4	**16.8**	0.80	0.14	0.21	0.30	0.38	0.49	0.62	0.72	0.80	**0.86**	0.84	0.75
(g)	Pot	**16.9**	17.9	0.89	0.21	0.43	0.58	0.69	0.75	0.78	0.81	**0.81**	0.78	0.74	0.58
(h)	Person	25.8	**13.7**	0.84	0.54	0.19	0.26	0.31	0.36	0.43	0.48	0.52	0.57	**0.65**	0.61
(i)	Speaker	**20.8**	22.8	0.83	0.48	0.36	0.55	0.74	0.81	0.88	0.92	**0.94**	0.89	0.60	0.36
(j)	Bicycle	19.1	**10.8**	0.60	0.21	0.74	0.83	0.88	0.91	0.92	**0.93**	0.92	0.88	0.81	0.76
Mean		23.7	**15.9**	0.86	0.44	0.52	0.63	0.70	0.75	0.78	0.81	**0.81**	0.80	0.75	0.60

For experiment 2 we compare the labeling time in seconds (s), against the M-Ontomat-Annotizer [27]. For experiment 3 we compute the overlap ratio (O) with LabelMe segmentations [6] for 10 plays and report results for a varying number of pixel selections. For each experiment and each image region the best results are denoted in bold. The segmented image regions are summarized separately in Fig. 7

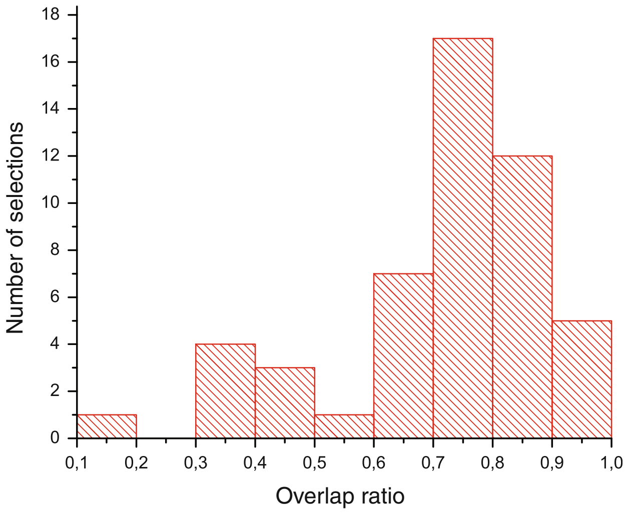


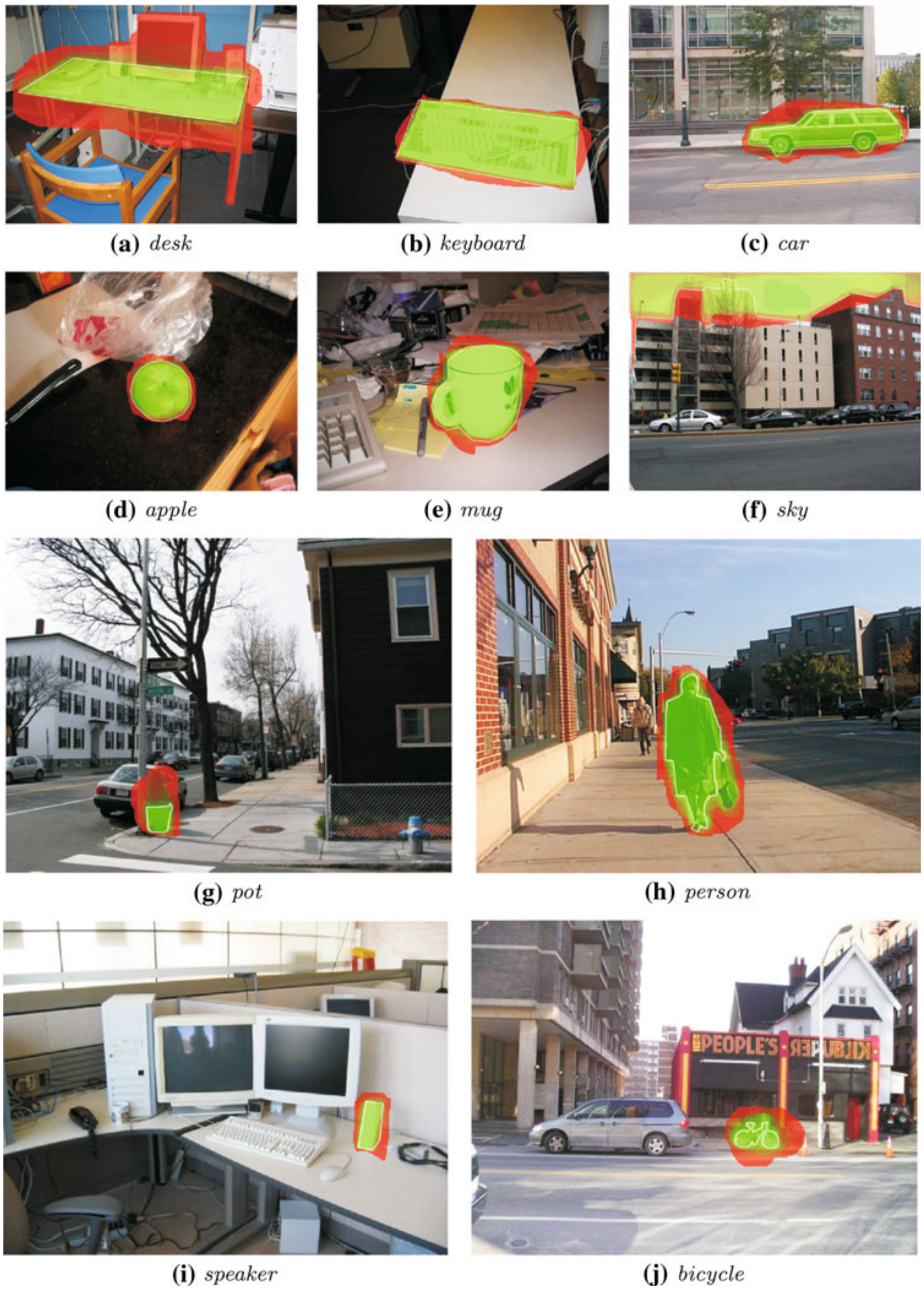
Fig. 6 Results for experiment 3: segmentation accuracy of 50 randomly chosen games. The histogram shows the distribution of the overlap ratios. The results indicate that, overall, the segmentations after one game are reasonably close to the LabelMe segmentations

In the second part of the experiment, we have inspected the number of times a pixel was selected. We summarize the quantitative results in Table 3 and visualize qualitative results of selection combinations in Fig. 7. On average, pixels selected minimally 6 or 7 times give the best accuracy with an overlap ratio of 0.81. A minimum of 8 pixels is a close runner-up and gives an average of 0.80. However, when looking at the images individually, the best selections are spread over 4, 6, 7, 8, 9 and 10 minimally selected pixels. In 5 cases, with ‘speaker’, ‘desk’, ‘car’, ‘bicycle’ and ‘sky’, the best combination selection was equal or better than the single best selection. The best combination selections are always better than the single worst selection. However, not all combination selections are better than the single best selection; implying that much is to be gained from a mechanism able to predict whether to fuse or not to fuse individual segmentations.

6 Conclusion

We have investigated which interactive multimedia system components should be present in an annotation tool to enable the quick labeling of several objects in images, in a semantically and region-based manner (see Table 1). To resolve the flaws of existing annotation tools we propose the Name-It-Game, a prototype annotation game allowing for semantic labeling of image regions in a playful manner. We consider an added semantic structure, by means of the WordNet ontology, the main innovation of the Name-It-Game.

Our experiments confirm the effectiveness and efficiency of the Name-It-Game. In experiment 1 we showed that adding semantics to existing image labeling games, results in less abstract and therefore better guessable words. In addition, by connecting labels in an ontology, the new possibilities of guessing using synonyms and supporting the guesser with valuable hints, speeds up the game and could potentially make it more fun to play. Last but not least, by including the user-provided labels in an ontology we obtain a structured and more rich image-region description with potential for automated reasoning. Experiment 2 shows that the Name-It-Game accelerates the labeling process compared to desktop annotation tools with the same components. The few actions required to annotate



(a) *desk* (b) *keyboard* (c) *car* (d) *apple* (e) *mug* (f) *sky* (g) *pot* (h) *person* (i) *speaker* (j) *bicycle*

Fig. 7 Results for experiment 3: segmentation accuracy. The *pure red* pixels are selected only once and the *pure green* pixels are selected ten times. Everything in between is rendered by using a gradient from red to green. The *white line* shows the LabelMe selection (rendering effect only visible in online color version). For quantitative results, see Table 3

with the Name-It-Game is what sets it apart from other annotation tools. As shown in experiment 3, the accuracy of the segmentations after one play is reasonably good compared to carefully created selections by experts. However, a good segmentation cannot be guaranteed after one play. The size of the object, the shape, the player, and possibly the scoring system can affect the quality of the segmentation. An algorithm taking these factors into account jointly, could quite possibly help us decide which segmentations are good, and which are not, in future releases of the Name-It-Game.

Arguably, the best parameters for evaluating a game are not so much the efficiency or the accuracy; but the playability, affective issues, and the user satisfaction. We consider these human-centered factors the most important improvements for future versions of the Name-It-Game, which can make the game more fun and possibly results in even better image-region annotations.

Acknowledgments This work is supported by the STW SEARCHER project.

References

1. Hanbury, A.: A survey of methods for image annotation. J. Vis. Lang. Comput. **19**(5), 617–627 (2008)
2. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: ideas, influences, and trends of the new age. ACM Comput. Surv. **40**(65), 1–60 (2008)
3. Enser, P.: Visual image retrieval: seeking the alliance of concept-based and content-based paradigms. J. Inf. Sci. **26**(4), 199–210 (2000)
4. Golder, S.A., Huberman, B.A.: The structure of collaborative tagging systems. J. Inf. Sci. **32**(2), 198–208 (2006)
5. Macgregor, G., McCulloch, E.: Collaborative tagging as a knowledge organisation and resource discovery tool. Libr. Rev. **55**(5), 291–300 (2006)
6. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. Int. J. Comput. Vis. **77**(1–3), (2008)
7. Barnard, K., Fan, Q., Swaminathan, R., Hoogs, A., Collins, R., Rondot, P., Kaufhold, J.: Evaluation of localized semantics: data, methodology, and experiments. Int. J. Comput. Vis. **77**(1–3), 199–217 (2008)
8. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
9. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: a high-definition ground truth database. Pattern Recognit. Lett. **30**(2), 88–97 (2009)
10. Naphade, M.R., Smith, J.R., Tešić, J., Chang, S.-F., Hsu, W., Kennedy, L.S., Hauptmann, A.G., Curtis, J.: Large-scale concept ontology for multimedia. IEEE MultiMed. **13**(3), 86–91 (2006)
11. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proceedings IEEE Computer Vision and Pattern Recognition (2009)
12. Chua, T.-S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.-T.: NUS-WIDE: A real-world web image database from National University of Singapore. In: Proceedings ACM International Conference on Image and Video Retrieval (2009)
13. Fellbaum, C. (ed): WordNet: an electronic lexical database. The MIT Press, Cambridge, USA (1998)
14. Hollink, L., Schreiber, G., Wielemaker, J., Wielinga, B.: Semantic annotation of image collections. In: Proceedings international conference for Knowledge Capture Workshop on Knowledge Markup and Semantic Annotation (2003)
15. Hyvönen, E., Styrman, A., Saarela, S.: Ontology-based image retrieval. In: Proceedings XML Finland conference, pp. 15–27 (2002)
16. Gao, Y., Fan, J.: Incorporating concept ontology to enable probabilistic concept reasoning for multi-level image annotation. In: Proceedings ACM International Workshop on Multimedia Information Retrieval, pp. 79–88 (2006)
17. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: Proceedings SIGCHI Conference on Human Factors in Computing Systems, pp. 319–326 (2004)
18. von Ahn, L., Liu, R., Blum, M.: Peekaboom: a game for locating objects in images. In: Proceedings SIGCHI conference on Human Factors in Computing Systems, pp. 55–64 (2006)
19. Squigl, http://www.gwap.com
20. Turnbull, D., Liu, R., Barrington, L., Lanckriet, G.: A game-based approach for collecting semantic annotations of music. In: Proceedings International Conference on Music Information Retrieval (2007)
21. Law, E.L.M., von Ahn, L., Dannenberg, R.B., Crawford, M.: TagATune: A game for music and sound annotation. In: Proceedings International Conference on Music Information Retrieval (2007)
22. van Zwol, R., Garcia, L., Ramirez, G., Sigurbjörnsson, B., Labad, M.: Video tag game. In: Proceedings International World Wide Web Conference (2008)
23. Gligorov, R., Baltussen, L.B., van Ossenbruggen, J., Aroyo, L., Brinkerink, M., Oomen, J., van Ees, A.: Towards integration of end-user tags with professional annotations. In: Proceedings International Web Science Conference (2010)
24. Gonçalves, D., Jesus, R., Grangeiro, F., Romao, T., Correia, N.: Tag around: a 3D gesture game for image annotation. In: Proceedings International Conference on Advances in Computer Entertainment Technology, pp. 259–262 (2008)
25. Seneviratne, L., Izquierdo, E.: An interactive framework for image annotation through gaming. In: Proceedings ACM International Conference on Multimedia Information Retrieval, pp. 517–526 (2010)
26. Yao, B., Yang, X., Zhu, S.-C.: Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks. In: Energy Minimization Methods in Computer Vision and Pattern Recognition, vol. 4679, LNCS, pp. 169–183, Springer (2007)
27. Petridis, K., Anastasopoulos, D., Saathoff, C., Timmermann, N., Kompatsiaris, Y., Staab, S.: M-ontomat-annotizer: Image annotation linking ontologies and multimedia low-level features. In: B. Gabrys, R.J. Howlett, and L.C. Jain, editors, KES (3), vol. 4253 of LNCS, pp. 633–640, Springer (2006)
28. Halaschek-Wiener, C., Golbeck, J., Schain, A., Grove, M., Parsia, B., Hendler, J.: Photostuff—an image annotation tool for the semantic web. In: Proceedings International Semantic Web Conference (2005)
29. Hollink, L., Nguyen, G., Schreiber, G., Wielemaker, J., Wielinga, B., Worring, M.: Adding spatial semantics to image annotations. In: Proceedings International Workshop on Knowledge Markup and Semantic Annotation at ISWC (2004)
30. Flickr, http://www.flickr.com
31. Volkmer, T., Smith, J.R., Natsev, A.(P.): A web-based system for collaborative annotation of large image and video collections: an evaluation and user study. In: Proceedings ACM international conference on Multimedia, pp. 892–901 (2005)
32. Marlow, C., Naaman, M., Boyd, D., Davis, M.: Ht06, tagging paper, taxonomy, flickr, academic article, to read. In: Proceedings International Conference on Hypertext and Hypermedia, pp. 31–40 (2006)
33. Ames, M., Naaman, M.: Why we tag: Motivations for annotation in mobile and online media. In: Proceedings SIGCHI Conference on Human Factors in Computing Systems, pp. 971–980 (2007)
34. Sorokin, A., Forsyth, D.: Utility data annotation with amazon mechanical turk. In: Proceedings IEEE Computer Vision and Pattern Recognition Workshops (2008)