

Multiscale Annotation of Still Images with GAT

Xavier Giro-i-Nieto and Manuel Martos
Technical University of Catalonia (UPC)
Barcelona, Catalonia / Spain
xavier.giro@upc.edu

ABSTRACT

This paper presents GAT, a Graphical Annotation Tool for still images that works both at the global and local scales. This interface has been designed to assist users in the annotation of images with relation to the semantic classes described in an ontology. Positive, negative and neutral labels can be assigned to both the whole images or parts of them. The user interface is capable of exploiting segmentation data to assist in the selection of objects. Moreover, the annotation capabilities are complemented with additional functionalities that allow the creation and evaluation of an image classifier. The implemented Java source code is published under a free software license.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User interface

General Terms

Design

Keywords

interactive, segmentation, annotation, semantics, image

1. MOTIVATION

The large and growing amount of visual digital data acquired nowadays has raised the interest for systems capable of its automatic analysis from a semantic point of view. After a first generation of algorithms in which specific-case solutions were developed through an expert study of the problem (eg. text or face recognition), it is a general trend in the computer vision community to try to develop generic solutions that can be easily adapted to a diversity of domains. Pattern recognition techniques have been successfully applied to a broad range of applications in computer vision, especially in their supervised learning variant. This type

of problems usually works with images and videos that significantly represent the problem that is to be solved. This dataset is split in two parts: a first one to train a classifier and a second one to evaluate the expected performance of the learnt model. In order to perform both tasks, it is necessary to previously annotate the dataset, a task that requires some kind of human interaction, whether explicit or implicitly.

Before training a classifier, pattern recognition problems require the extraction of features that map images into a space where decision boundaries can be estimated. Good features are those that confine the instances of each class to a portion of the feature space that does not overlap with the instances associated to the rest of the classes. In the case of image analysis, a first solution is to use features extracted after considering images at the *global* scale. This approach simplifies the manual annotation task as the expert only needs to decide whether the image represents or contains an instance of the target class. However, in those cases where instances appear in a specific part of the image, like in object detection problems, global scale annotation makes it more difficult to train good classifiers, as they need to discriminate which portions of the positively annotated images are actually related to the modelled class. In these situations, a *local* scale annotation provides better features for the classifier at the expense of a higher effort from the annotator, who must manually indicate the area of support of the instance. This task requires the introduction of a graphical user interface to assist users into the determination of these areas.

The annotation process does not only require selecting visual data but also associating it to a semantic class. If this class has a semantic meaning, as in most computer vision tasks, these semantics must be defined in an additional data structure. Ontologies are the most common solutions adopted by the scientific community as they define classes in a formal and structured manner. Successful computer vision techniques not only base their results on the signal processing algorithms but also on semantic reasoning processed at a higher level. The use of ontologies introduces context in the analysis task and offers an opportunity to fuse image analysis with other modalities such as text and audio. For these reasons, annotation tools not only need to offer a workspace to select images and regions but must also provide mechanisms to handle ontologies.

This paper extends a previous work [2] where GAT (Graphical Annotation Tool) was introduced for the annotation of still images at the local scale. This original version has been

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

VIGTA '12 Capri, Italy

Copyright 2012 ACM ISBN 978-1-4503-1405-3 ...\$10.00.

improved with an integrated environment where annotations can be generated at both global and local scales. This core functionality has been complemented with a new perspective to train and evaluate image classifiers. GAT is addressed to an academic audience that can find in this software a solution to generate a ground truth of MPEG-7/XML standard annotations, which can be later used to test their own classification algorithms.

The rest of the paper is structured as follows. Section 2 reviews some of the related work in the field of semantic annotation of still images, both at the local and global scales. Section 3 presents the basic workflow with GAT, an overview of the different parts that are described in the remain of the paper. 4 presents the different options to select areas of support at the global and local scales. Section 5 describes how semantic data is displayed while Section 6 explains how an image classifier can be trained and evaluated within the same tool. Section 7 explains the intended architecture and used data formats and, finally, Section 8 draws the conclusions and provides instructions about how to download and test this tool.

2. RELATED WORK

The manual annotation of images is a time-consuming task that has been an intense research area for the last decade [1] [4]. There exist a variety of solutions that have explored topics such as crowd-sourcing, usability, interactive segmentation and ontology management.

At the global scale, the TRECVID evaluation campaign used the IBM Efficient Video Annotation (EVA) tool [11] to annotate the presence of a certain concepts in video shots. This web-based tool painted the box around the video keyframes with one color (green, red or white) to visually code the associated label (positive, negative or neutral). The user could change the initial red frame assigned by default by clicking on the keyframes. This code of colors has been adopted in this work to indicate the labels at the global scale, although the selection mechanism has been modified to provide more flexibility to the user. At the local scale, an online interface was developed by the LabelMe project [8] to collect a large amount of object silhouettes. Users drew a polygon around the object, which provided a local but somewhat rough annotation of it. The user also introduced a free textual label that was mapped onto the WordNet ontology.

A popular strategy for obtaining crowd-sourced annotations is through online games. The Extra Sensory Perception (ESP) game [12] collected textual labels at the global scale by showing to a pair of players the same image. Players were prompted to enter words related to the shown image and, when an agreement was obtained between different players, they were rewarded with points. The label was considered correct by the authors when different pairs agreed on a word. This idea was extended to the local scale in the Name-It-Game [10], where objects were outlined by a *revealer* player and had to be predicted by a second *guesser* player upon a gradual appearance of the selected object. This interface combined freehand and polygonal segmentations, and the considered concepts were extracted from the WordNet ontology.

The main drawback of web-based tools and games is that they need setting up a server, a task that may require advanced technical skills. Although this architecture is appropriate for a collaborative annotation effort, it poses prob-

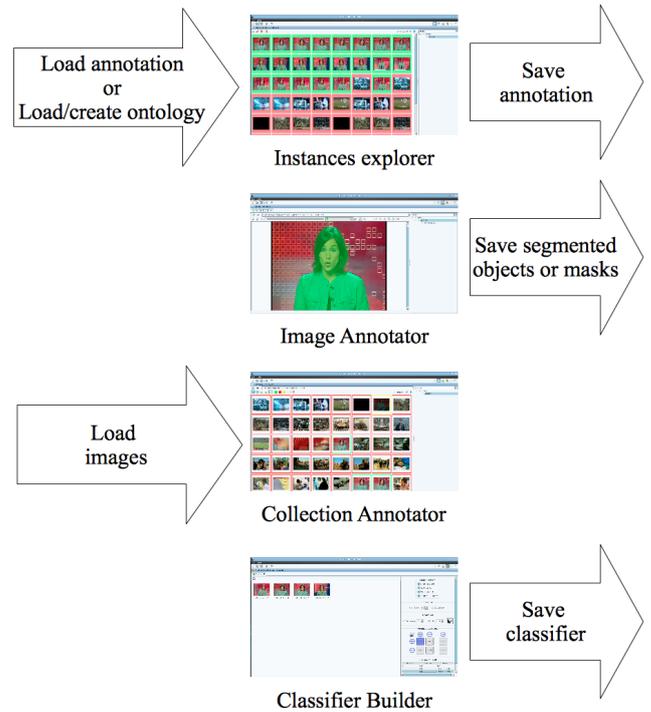


Figure 1: The four perspectives in GAT.

lems when simpler configurations are preferred. GAT has been developed as a multi-platform desktop tool to facilitate its adaptation from third part users. However, the source code is also prepared to work with a remote repository, as reported in [3].

There exist other desktop solutions apart from GAT. M-OntoMat-Annotizer [7] is a region-based annotation tool that combines multimedia and domain-specific ontologies. This software contains a segmentation engine that lets users associate concepts to selected sets of region. The tool is also capable to extract low level visual descriptors and generate MPEG-7 descriptions that contain both perceptual and semantic information. The MPEG-7 data format has also been adopted by GAT, as it offers a formal language to represent content at both low and high level. However, this tool provides a single interface for both global and local annotations, and it requires an individual processing of each image. GAT facilitates the annotation at the global scale, with a dedicated perspective based on thumbnails and selection tools for the fast labelling of images.

3. WORKFLOW

GAT provides four different perspectives aimed at guiding the user during the different stages of the annotation. Figure 1 offers an overview of them as well as the input and output data associated to each of them. The user can jump at any moment from one perspective to another through dedicated icons located in the toolbar.

After launching GAT, the *Instances explorer* is presented. This perspective allows a quick overview of the instances already annotated so, at launch time, it will appear empty. At this point the user can whether load an annotation previously saved in disk or select an ontology to be associated

to a new annotation. In the later case, a floating window will appear prompting the user with three possible options: exploring the file system to load an existing ontology, read the ontology from a remote URL or creating a new one from scratch. The last option will show a new panel with a simple ontology editor, where classes can be added, removed and rename. This editor can be accessed again in the future during the annotation. Any new ontology must be saved in a file so that new annotations can refer to it.

Once the annotation is initialized, the next stage corresponds to the visual labelling of images. This stage requires changing to the *Collection Annotator* perspective. This perspective is populated with the thumbnails of the images selected by the user from a local directory. The user can directly label images at the global scale from this perspective (presented in Section 4.1), or can double click on any of the thumbnails to generate a local annotation of the image (explained in Section 4.2). This second action will change to the *Image Annotator* perspective, where the selected image occupies the main panel.

The annotated instances can always be reviewed by returning the *Instances explorer*, that contains a *disk* icon to save the annotation to a local file. This perspective is also the entry point to the *Classification* perspective, where the annotated images are used to train an image classifier. GAT offers the necessary tools to set up a cross validation experiment and analyse the results both numerically and visually. From this perspective, the user can also export the trained classifier for its external exploitation.

4. VISUAL LABELLING

The annotation of images can be performed at two basic visual scales: global or local. In the global case the area of support is the full image, while local annotations mark a subset of the image pixels that depict a semantic object. GAT provides different tools to assist users for a quick interaction with images at both scales.

All presented strategies share a basic workflow for annotation. Firstly, the user selects the segments of images that are to be annotated and, as a response, the interface clearly highlights the selection. At this point, the user can decide to modify the selection or validate it with a right-click on the mouse. After validation, the new instance(s) is added to the current annotation and clearly marked on the interface. This way, the right-click becomes the common action for validation.

4.1 Global scale

Annotations at the global scale normally consider several images. GAT provides a dedicated *Collection* perspective that explores the content of a folder in the file system and shows the thumbnails of the included images. In most cases, viewing the thumbnails is enough for users to decide about the label but, if necessary, a double click on any of them displays the full image on a new *Image* tab.

A broad range of machine learning techniques require that annotations consider not only which observations correspond to a semantic class but also which of them do not correspond to the class. A classic example are binary classifiers, that use two types of labels: positive and negative. In some situations a third type of label, the neutral one, is also used. This label just states the existence of the observation. These neutral images are usually discarded for training or experi-



Figure 2: Selected vs annotated images.

mentation [11] as its inclusion may harm the overall performance. These three types of labels are supported in GAT only in the case of global annotations, as local annotations usually imply a positive label for the selected segment and a negative label for the rest of the image.

The assignment of global labels starts by clicking on one of the six icons located on the perspective’s toolbar. Their color intuitively indicates what label are they related to: green (positive), red (negative) or yellow (neutral). These icons provide two different types of selection tools: individual or all. The first group activates the associated label so that every new click will associate the label to the image. The second group sets the selected labels to all currently non-annotated images. For example, this functionality becomes very practical in those cases where only a few of the displayed images belong to the class. In this situation, an initial red labelling to all thumbnails can be later be corrected by switching the appropriate thumbnails to green.

Figure 2 shows how selected and annotated thumbnails are distinguished. When a thumbnail is selected, a frame of the associated label’s color is painted around the panel containing the thumbnail. When the assigned labels are validated with a right-click, the thumbnail panel is painted with the color of the label.

4.2 Local scale

As previously explained, a double-click on a thumbnail of the *Collection Annotator* perspective will activate the *Image Annotator* perspective, where the selected image is shown in a newly created tab. Apart from providing a more detailed view of the image, this tab allows its local annotation. All local annotations are assigned to the positive label so, in this mode the color code used for global annotations does not apply. The color of the markers used for local selection can be configured by the user to avoid visual confusion between the instance selection and the background.

Local-scale solutions can be divided in two groups depending on the sought precision. A first family of techniques provides *rough* descriptions of the objects [8] [10], giving approximate information about their location and shape, normally, using geometric figures. A second option for local annotations is the precise *segmentation* of those pixels that represent the object, by defining the exact area of support associated to the object [7]. GAT provides tools for both options, with special emphasis on interactive segmentation strategies for the second case.

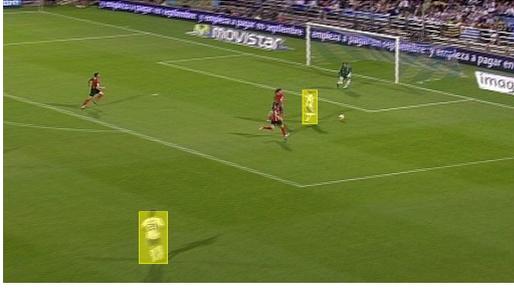


Figure 3: Rough annotation of soccer players.

4.2.1 Rough annotation

GAT allows drawing geometric markers over the image to indicate the local presence of a semantic instance. The catalogue includes points, lines and rectangles, which can be combined in the same image. Figure 3 shows an example of a rectangle-based rough annotation of two soccer players.

4.2.2 Interactive segmentation

Systems offering precise local annotations can be classified into region-based or contour-based approaches. Region-based annotations let the user select among a set of segments from an automatically generated partition of the image, while contour-based solutions aim at generating a curve that adjusts to the pixels located at the border between object and background. GAT provides four methodologies based on the first family to interactively generate a segmentation of the instance. In all of them, the success of the interaction is tightly dependent on the goodness of the segmentation. GAT does not include a segmentation engine but several state of the art techniques offer nowadays enough precision to be used into the proposed interactive framework [5] [6].

Partition-based technique.

The proposed partition-based technique requires a previous segmentation of the image in regions. This selection mode requires the user to draw a rectangle around the instance so that the algorithm automatically selects the partition regions which are completely included in the rectangle. The selected regions are shown to the user as transparent in an overlaid mask, as shown in Figure 4. The user can modify the suggested result if unexpected regions were selected or if some of the expected regions were not selected. A left-click on the image will be mapped to a region in the partition and its selection state switched. This strategy is very intuitive for users, who are very familiar with drawing rectangles and clicking.

Partition Tree-based techniques.

In addition to the initial segmentation required by the partition-based technique, the three proposed solutions in this section require the creation of a hierarchical structure by iteratively merging the most similar neighbouring regions. The resulting structure is a *Partition Tree (PT)* which, in the specific case of merging two regions at every iteration, is named *Binary Partition Tree (BPT)* [9]. Figure 5 shows the hierarchical decomposition of an image into the regions defined



Figure 4: Rectangle marker and selected regions.

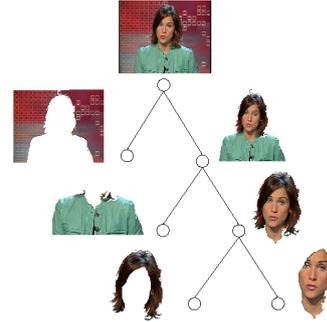


Figure 5: Binary Partition Tree.

by a BPT, the data structure considered in this work.

The first application of BPTs to interactive segmentation is the propagation of labels through its structure. In this case, the user interaction requires drawing *scribbles* on the image specifying if these markers are on the object or on the background. Every time a new scribble is added, a subset of BPT leaves are also labelled as object or background. Object labels are iteratively propagated to the parent node in the BPT if the subtree defined by the considered node's brother contains at least one object label, but no background label. Similarly to the rectangle and points scheme, the selection can be refined through successive iterations. Figure 6 shows a first step (a) where an object scribble (green) is drawn over a face. Step (b) shows how the label propagation has erroneously selected some regions belonging to the background, so a background (red) scribble is drawn over them to finally obtain a better segmentation in step (c).

The two other BPT-based modes refer to navigation through the tree structure in order to select the nodes representing the object. The main difference between the two navigation modes is whether an initial click is needed to start the



Figure 6: Sequential segmentation with scribbles.

selection.

The *clickable* mode starts with a left-click on the area of support of the object. With this action, the user is implicitly selecting one branch from the PT, as every pixel in the image corresponds to one, and only one, branch in the PT. After this first user interaction, the interface highlights the region associated to the PT leaf so that the user can evaluate if the proposed region correctly depicts the object. If this is not the case, the selected PT node can be modified by rotating the mouse wheel, moving upwards or downwards in the branch at every wheel rotation. Every new move will expand or contract the selection depending on the direction of the rotation. The navigation path is defined between the PT root, where the whole image is selected, and a PT leaf, where a region at the initial partition is shown. A second left-click will save the currently selected node and allow choosing regions from other PT branches before the final validation with a right click.

The *clickless* mode is based on the same principles as the *clickable* case but it requires less interaction from the user side. The multi-scale navigation and multiple branch selection are shared features among them, but in the *clickless* mode PT branches are selected by just placing the cursor over a region, with no need of an initial click. This means that, whenever the cursor is over the image panel, a region is highlighted. Some users find this option too confusing due to the high activity on the panel. For this reason, GAT offers the two options and lets users decide which of them better suits their preferences.

The reader is referred to [2] for a more accurate description of the navigation scheme through the PT.

5. SEMANTIC PANEL

The presented perspectives always contain a *Semantic Panel* located on the right-side of the interface. This panel includes a tree whose root corresponds to the name of the ontology and its children the semantic classes available for annotation.

In the *Instances Explorer* perspective, a click on a class node will show in the main panel all the images annotated for the class. This operation allows reviewing the annotation and deleting those instances that might have been wrongly created.

In the *Collection Annotator* perspective, the behaviour is slightly different, as the presented thumbnails are associated to the current directory, so the selection of the class will just highlight those listed images with a label associated to the class. Moreover, the selected class also indicates the reference of the *positive / negative / neutral* labels added during the annotation.

In the case of the *Image Annotator* perspective, the semantic tree is expanded in an additional level, adding one node for each instance to every class node. When the user selects one of this instance nodes, the related local annotation is shown on the main panel. A click on a class node will show all instances of the class, and a click on the root will display all annotated instances in the image. Analogously to the *Collection Annotator* perspective, the selection on the tree also indicates the class that is being annotated. The whole interface is shown in Figure 4, where the Semantic Panel highlights one instance of the semantic class “Anchor”.

6. EVALUATION OF CLASSIFIERS

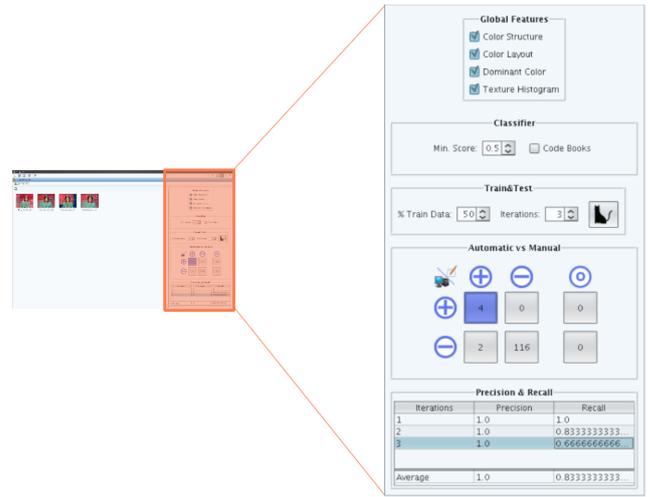


Figure 7: Classification perspective.

In addition to all the tools for annotation, GAT includes a perspective that exploits the generated annotation in the framework of an image classification system. This perspective provides an intuitive environment to evaluate an image classifier trained with the annotated content.

The *Classification* perspective is accessible by clicking an icon on the toolbar of the *Instances explorer* perspective. This action switches perspectives and creates a new tab associated to the selected class, as shown in Figure 7. The tabs in the *Classification* perspective are organized in two large areas: a central panel that shows image thumbnails and a vertical panel on the right to control the parameters for the classification and the evaluation.

The control panel allows the selection of different parameters related to an image classification engine. In particular, it allows choosing among a catalogue of visual descriptors, setting a minimum confidence score for detection and deciding if a codebook must be used during the process. A second type of controls refer to the evaluation process itself. The adopted approach follows a cross-validation scheme with a random partition between training and test data. The user can select the amount of folds to run as well as the proportion of annotated images assigned to the training and test sets.

A left-click on the cat-shaped icon launches the evaluation process. In each iteration of the cross-validation process, the dataset is partitioned and the training data is used to learn the visual model for the class. Once built, the images from the test partition are classified one by one as belonging to the class or not. The label predicted by the classifier is compared with the annotated ground truth, so that the every test image is counted as a true or false classification.

The graphical interface allows a rapid assessment of the results. Firstly, the panel on the right includes a table that displays the precision and recall obtained on each iteration of the cross-validation. The last row of the iterations table averages the precision and recalls obtained in each cross-validation fold. The user can click on any row of that table, an action that selects the data to be displayed in the main panel of thumbnails. The images shown there depend on the

active button from another grid panel, that represents the confusion matrix. The diagonal of the matrix corresponds to the correct predictions, while the rest of cells in this grid corresponds by errors from the classifier. Given the single-class nature of the perspective, the size of the square grid is 2x2, each of its cells associated to a *true/false positive/negative* prediction. There exists though an additional column that corresponds to the neutral labels. Whenever the user clicks on any on these cells, the large panels of thumbnails is refreshed by showing the images that correspond to the set.

The *Classification* perspective also allows exporting a model of the selected class to any location in the file system. This way, if the user is satisfied with the presented results, a version of the classifier can be saved for its external exploitation. In that case, a new model is built considering all annotated images as belonging to the training dataset.

7. DATA FORMATS

Regarding data formats, GAT is based on MPEG-7/XML to code the ontologies, annotations and BPTs. Examples of all types of documents are provided with the software package. The most common image coding formats are also supported by GAT, as it uses the native Java classes. In addition, it also supports a developed PRL data format to code image partitions of 32 bits per pixel. Nevertheless, partitions can be coded in any other format supported by Java (PNG, BMP,...).

GAT is designed to both read precomputed BPTs or use an external tool to use them whenever need. In the second case, this additional tool can be a binary file in the local machine or a web service accessed through the Internet.

8. CONCLUSIONS

This paper has presented GAT, a tool designed for the semantic annotation of images at the local and global scale. This work is addressed to researchers in the computer vision and semantic fields who want to manage images in an intuitive framework. This project has been funded by two industrial companies who agreed to open the source code of this tool under a free software license to facilitate its promotion, reuse and further extension among the scientific community. The source code is available on a public website¹, where video-demos of the software can be watched and the tool itself downloaded and launched.

GAT is currently being used in a teaching environment for a practical exercise on image classification, where university students complete the whole annotation, training and evaluation cycle with an intuitive and graphical environment. Moreover, it has been used to annotate datasets of hundreds of images at the local scale, which have been exploited by object recognition engines.

Future work will concentrate on an evaluation of the system following the guidelines suggested in [5]. The experiments will evaluate both the quality of the generated data as well as the time invested by the users to generate them.

¹<http://upseek.upc.edu/gat/>

9. ACKNOWLEDGMENTS

This work was partially funded by the Catalan Broadcasting Corporation through the Spanish project CENIT-2009-1026 BuscaMedia, and by Spanish project TEC2010-18094 MuViPro: "Multicamera Video Processing using Scene Information: Applications to Sports Events, Visual Interaction and 3DTV".

10. REFERENCES

- [1] S. Dasiopoulou, E. Giannakidou, G. Litos, P. Malasioti, and Y. Kompatsiaris. A survey of semantic image and video annotation tools. In *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution*, volume 6050 of *Lecture Notes in Computer Science*, pages 196–239. Springer Berlin / Heidelberg, 2011.
- [2] X. Giro-i Nieto, N. Camps, and F. Marques. Gat, a graphical annotation tool for semantic regions. *Multimedia Tools and Applications*, 46(2):155–174, 2010.
- [3] X. Giro-i Nieto, C. Ventura, J. Pont-Tuset, S. Cortes, and F. Marques. System architecture of a web service for content-based image retrieval. In *Proc. ACM Intl' Conference on Image and Video Retrieval, CIVR '10*, pages 358–365, 2010.
- [4] A. Hanbury. A survey of methods for image annotation. *Journal of Visual Languages and Computing*, 19(5):617 – 627, 2008.
- [5] K. McGuinness and N. E. O'Connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434 – 444, 2010.
- [6] A. Noma, A. B. Graciano, R. M. Cesar, L. A. Consularo, and I. Bloch. Interactive image segmentation by matching attributed relational graphs. *Pattern Recognition*, 45(3):1159 – 1179, 2012.
- [7] K. Petridis, D. Anastasopoulos, C. Saathoff, Y. Kompatsiaris, and S. Staab. Montomat-annotizer: Image annotation, linking ontologies and multimedia low-level features. In *Intl. Conf. on Knowledge Based, Intelligent Information and Engineering Systems*, 2006.
- [8] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3):157–173, 2008.
- [9] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *Image Processing, IEEE Transactions on*, 2000.
- [10] J. Steggink and C. Snoek. Adding semantics to image-region annotations with the name-it-game. *Multimedia Systems*, 17:367–378, 2011. 10.1007/s00530-010-0220-y.
- [11] T. Volkmer, J. R. Smith, and A. P. Natsev. A web-based system for collaborative annotation of large image and video collections: an evaluation and user study. In *13th annual ACM Intl' Conference on Multimedia*, pages 892–901, 2005.
- [12] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '04*, pages 319–326, 2004.