An Integrator Tool for Object Class Recognition in Digital Imagery

Vinícius A. De Melo
GSORT - Instituto Federal da Bahia
Rua Emídio Santos, s/n,
Salvador - BA
Email: stdcoutvinicius@gmail.com

Manoel C.M. Neto
GSORT - Instituto Federal da Bahia
Rua Emídio Santos, s/n,
Salvador - BA
Email: manoelnetom@gmail.com

***Abstract*—The use of video cameras in different devices and applications generates a huge amount of data, which in most cases are only consumed by humans. One of the key points to allow these data to be processed by computer systems is the detection and recognition of different objects that can be part of an image. This allows, for example, not only to use a camera as a device that produces videos or images but also as a sensor that can be used in the development of ubiquitous applications. Thus, the object class recognition task on digital imagery can be seen as a support tool for ubiquitous computing.**

This paper presents a tool for integrating object detection solutions per domain, guided by a systematic literature review aiming to expose the state-of-the-art of the object class recognition task. The priority in this study was to select the most cited papers and used techniques. We extracted the results, showing the techniques used taking into account its performance, as well as its limitations and the challenges of the research area as a whole. From the results found, it was possible to point out the most used techniques in the area,the difficulties in its implementation due to the amount of detail, and how a tool that encapsulate these details can be useful for non-expert developers in computer vision.

***Index Terms*—Multimedia, Computer Vision, Detection, Object Classes, Tool, Integrator, CBIR, SLR**

I. Introduction

The Computer Vision emerged in order to reproduce the human visual ability to recognize three-dimensional objects and embed them into robots. By the time of its appearance in the 70s, the area called Computer Vision was differentiated from Digital Image Processing area, and was defined by the idea of recovering the three-dimensional structure of the real world as prior step to the complete semantic understanding of the scene. Researchers in this area then has been developing algorithms and mathematical mechanisms to detect objects in sets of images [1]. There are several types of problems in this area, such as: visual tracking in unconstrained environments, focus on pedestrians and vehicles with part-based detection, detection and description of events and activities from videos, etc. This paper focus on automatic detection and recognition of real-world objects in images [2].

Object recognition in images is a common challenge in computer vision. It can be divided into object instances recognition and object class recognition [2]. Object instances recognition consists on identifying an specific object known in advance. Object class recognition consists on identifying a class in which an object belongs [2]. The range of variations in colour, texture and shape within a class of object is a challenge for recognition. In this case, the problem is classified as intra-class variation [2]. However, we can assert that the identification is a more challenging task due the variety of classes that may have very similar visual characteristics (e.g. a tiger and a cat). This kind of problem is classified as inter-class variation.

The current accuracy of object class detection techniques can still be considered too low to be used in general purpose applications. So, this is still an open problem [2]. For example, in [3] the author highlights that there is not a technology sufficiently unified and matured to represent all the aspects of visual perception, and therefore suggest that each problem type should be approached by using a specific method. The same author considers that, in an image, we can not infer what is the class of objects it contains. He also says that, there is no way to know at what scale, position and orientation those objects will be placed and yet, under which light conditions or kind of scenario it will be inserted [3]. In this regard, Szeliski [1] indicates that even with all the advances, current computer vision systems are less capable than a two-year old child.

In [**?**], Smith affirms the importance of Computer Vision, as well as the use of sensors and RFID for ubiquitous systems. In [4], Computer Vision is indicated as the basis for developing applications with augmented reality. In his work Wirtz highlights the difficulty of implementing ubiquitous solutions with computer vision due to the current form of implementation, which is based on image transfer via Internet to remote processing. In the meantime, the DMCV (Direct Mobile Computer Vision) is presented as a solution based on processing in mobile devices using an *ad hoc* network with no need of Internet.

The major goal of the solution proposed here is to contribute to the area of ubiquitous systems, by simplifying the class object detection task by non-expert developers in computer vision. Those are some applications that can use objects class detection capabilities aimed here: detecting and counting animals on farms through unmanned aerial vehicle (UAV) equipped with camera, automobiles identification at crossings for automated control of lights, identification of nameplates vehicles in a way to settle debts verification by the competent traffic authority, etc.

The project here presents an integrative tool that encapsulates the details involved in detecting a certain class of objects individually. The aim is to reduce the problem related to the difficulty of using the object class recognition techniques. For each desired domain, the tool will provide a program that will perform the recognition tasks. The interface provided by the programs will encapsulate the complexity of algorithms and techniques involved. The developer will deal only with simple system calls as *detectObjects targetImage.png*.

This article is organizes as follows: Section 2 describes the systematic literature review performed, Section 3 presents the correlated works, Section 4 describes the proposed tool, Section 5 details two examples of object detection proposals from scratch, and Section 6 presents conclusions and final remarks.

II. Systematic Literature Review

Knowing and understanding what are the most important studies in this area is a key point to find solutions to the problems mentioned previously. In this context, this paper attempts to survey the state-of-the-art of object class recognition area by means of a Systematic Literature Review (SLR). This sectuibs is organized as follows: part A describes the method used to conduct this SLR, part B presents the results of this review, part C presents the analysis of these results, part D highlights some limitations of this work, and part E presents conclusions and final remarks.

A. Methodology for systematic review construction

The need to synthesize available research evidence created well established evidence-based disciplines such as medicine and education research method called systematic literature review [5], [6]. This practice has recently been recognized in several computing disciplines as, for example, software engineering and HCI [7], [8]. More recently, a new method derived from systematic literature reviews was introduced: systematic mapping studies. Such studies are more focused on developing classification schemes of a specific topic of interests and reporting on frequencies of publications which cover a given topic of the development classification schemes.

This work reports the findings of a study that was conducted by combining methods for systematic literature mapping and review to investigate the current state of research on object class recognition area. It is our goal to find the state-of-art of techniques, tools and methods used in this topic. For this, it is also intended to identify research issues unresolved in order to propose a project to present some contribution to this area. The details of research methods are described in the following subsections and were adapted from [9].

B. Study Design

This section presents the main focus, goals, highlights questions that this review attempts to answer and explains what research papers were included and excluded.

The focus of this literature review is based on Cooper's research outcomes, research methods, and practices or applications categories [10]. The research outcomes reveal gaps in the literature with regard to object class recognition. The findings are based on the systematic analysis of data collection of research material. The research methods are analyzed to provide an overview of approach evaluations used by researchers and their contribution focus. The focus on practices and applications shows useful information regarding what type of content is provided in prototypes, where they came from, and where they are presented.

Our goal is to integrate outcomes and synthesize the results. We also attempt to generalize findings across the collected research papers. For this, a survey was conducted on sources of relevant research in the area. These sources are those where convey the most important works in computer vision and computer science areas as a whole. We used keywords which describes the information of interest. Then the results were filtered in order to get only those that concerns to the area. For that, the paper's title, keywords and abstracts were analyzed. From this results we selected, the most cited ones, the newest and those that cites the most cited techniques for complete reading and contribution extraction.

Finally, two important questions to be answered by this review are:

1) What are the major techniques, methods and tools used for object class recognition?
2) What are the available tools to aid the object class recognition, including those that provide a high-level interface.

C. Data Collection

The first step on Data Collection phase consists of a database search through academic and state-of-the-art publication databases. This step also included a manual search in the proceedings of some of the main symposiums and conferences whose focus is object class recognition. Four digital libraries were identified to be systematic searched:

- ACM Portal (http://dl.acm.org/),
- IEEE Xplore (http://ieeexplore.ieee.org/),
- Google Scholar (http://scholar.google.com.br),
- Springer Link (http://link.springer.com).

These digital libraries were chosen because they:

1) own search engines that allow the use of logical expressions or equivalent mechanism,
2) include computer science publications or related topics that are related to the points being researched,
3) allow the search within metadata of publications, and
4) are accessible through the academic research network of the authors.

These digital libraries are commonly used as sources of systematic surveys in computer science research. Note that, not all digital libraries had the same features and search capabilities. So, it was necessary to apply modifications on the search for each specific library. Specially, Google Scholar although not being a publishing platform, presented results from other digital libraries and sources not mentioned here.

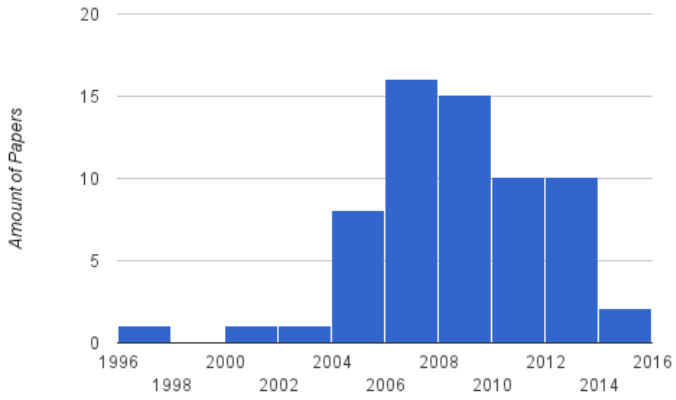

Figure 1. Publications per Year

For each source, we conducted a search that used Boolean expressions. The logical Boolean string used in the conducted search is listed below:

1) (“class recognition”) OR
2) ((“ class recognition"OR “class detection") AND “technique”) OR

To conduct a search using an equivalent string as above, it was necessary to understand each of the digital library’s advanced search features. The end result was that all papers retrieved had within their title, abstract, or keywords a combination of the keywords presented in the Boolean string. Before proceeding to the next phase, all duplicated publications were removed. The number of papers selected after this stage was 133.

Inclusion criteria were outlined in stage 2 to filter irrelevant studies from stage 1. The title and the abstract of each paper were individually examined for false positives. Thus, it was possible obtain a search result that contained all wanted keywords but without necessarily discussing the points of this review. At this point, a total number of 65 studies remained. In this group, were included the papers who met the inclusion criteria. These inclusion criteria are:

1) Papers that present and evaluate object class recognition methods.
2) Papers that present tools for object class recognition.
3) Papers that present techniques for object class recognition for a particular domain.
4) Papers that evaluates tools that provides a high level interface for object class recognition.

In stage 3, all the 65 remaining papers were read. With the the full text reading of each paper, it was possible to identify new papers that matched exclusion criteria, something that was not possible with the readings of title and abstract only. This stage was also utilized to extract data to be analyzed later. In the end, 20 papers remained. These papers include the most cited ones and those that address the top four object class recognition techniques.

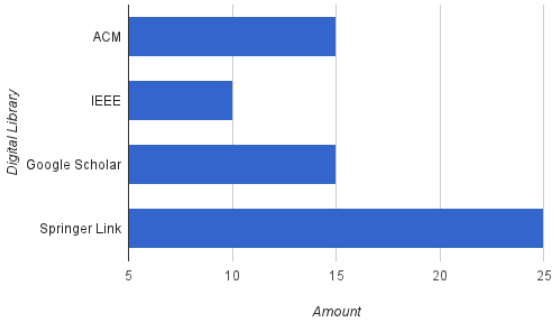

Figure 2. Papers per Library

D. Data Analysis

This section presents the data extracted from stage 3. A questionnaire was used to extract data from the literature in an iterative process. A first version of the questionnaire was designed and tested on a small subset of collected papers, revealing more variables that were brought to attention. After the refinement, the questionnaire was then used to extract data from all collected papers. A digital format for the questionnaire was utilized, using the Google Form[1] technology. The use of a digital questionnaire allowed to introduce new variables during the SLR. The questionnaire can be summarized as:

1) general Information for the paper:
   a) year of publication;

[1]http://www.google.com/drive/apps.html

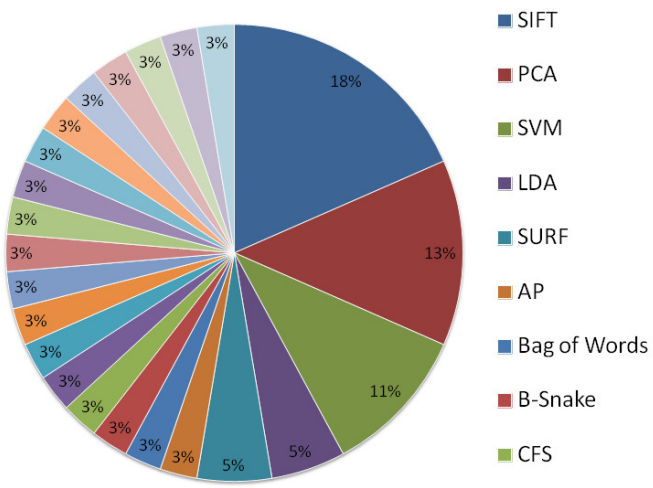

Figure 3. Techniques per Number of Citations

b) publication Source;

2) object class detection specific information:
   a) techniques for image matching;
   b) techniques domains;

III. Results

We selected 65 papers out of 133 items returned by search. The distribution of papers collected over the years is shown in Figure 1. The search returned papers dating from 1996 to 2014. The majority of articles have been published after 2005.

The articles selected in each platform were distributed as follows: Springer Link 25 papers (Springer was the library that returned more results), 15 papers in Google Scholar, 15 papers in ACM and 10 papers in IEEE. This distribution is presented in Figure 2.

The Figure 3 presents the most cited techniques for image matching. The top four techniques are SURF (Speed-ed Up Robust Features) with 5.3%, SVM (Support Vector Machine) with 10.5%, PCA (Principal Component Analysis) with 13.2% and SIFT (Scale-Invariant Feature Transform) with 18.4%. The Figure 4 shows the top 8 most cited papers and Figure 5 indicates the most cited domains.

A. Overview

In [11], the authors did a survey listing over than 300 contributions to the image processing area and other research sub-fields. This paper includes different research fields such as computer vision, machine learning, image recovery, human-machine interaction, database systems, data mining and web, information theory, statistics, psychology and other new fields that arise from the interaction between two or more fields, like machine learning and psychology.

The term CBIR (Content-Based Image Retrieval) was used in [11] and in [12] to describe any technology that aid an image classification by using their content as parameter. In this scenario, both papers identified the difficulty on the use of visual similarity for judging semantic similarity. The authors say that the human eye took a long time to evolve until reach the current precision stage. Therefore, to infer a function and pass it to a machine is still an open challenge. There is a semantic gap between a pure and simple visual content (low level) and a semantic context in which that content applies (high level) [11].

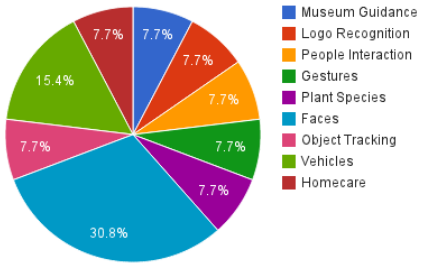

Figure 5. Domains per Amount of Citations

Still on paper [11], the authors understand as a sensorial gap, the difference between a real-world object and a discrete information stored on computer media. They point that distinct domains types can help to minimize this sensorial gap. These domains are necessary to interconnect sensorial and semantic gaps by means of visual attributes available, to meet to the user's demands. This process includes image processing and visual features definition.

Some challenges in object class recognition area can be summarized in two questions: How to represent images in order to allow search and to organize its contents? How to define similarity functions between image representations in a way that it could reflect the human perception? [12]. To answer these (and other) questions, [11] says that an image search engine should indicate features that reflects the user's intentions, thus making a reduction into the semantic gap. Some examples of search engines that already indicate these features are: QBIC [13], Pictoseek [14] and VisualSEEK [15]. To address the problem of human vision characterization in an algorithmic way, the authors in [11] suggest a trend in the use of statistical techniques and machine learning in CBIR. The automatic machine learning is typically used in clustering and classification, signature composition, similarity measures tuning and technical basis for search schemes.

Another shortcoming is the lack of methods and techniques that deal with segmentation. For example, the extraction of visual signature is a preliminary procedure for image analysis tasks and has the segmentation as a sub-task. In this context, there is an urgent need to develop a more effective framework for representing images contents and features extraction in order to avoid erroneous segmentation process, and also to enable automatic images annotation [16].

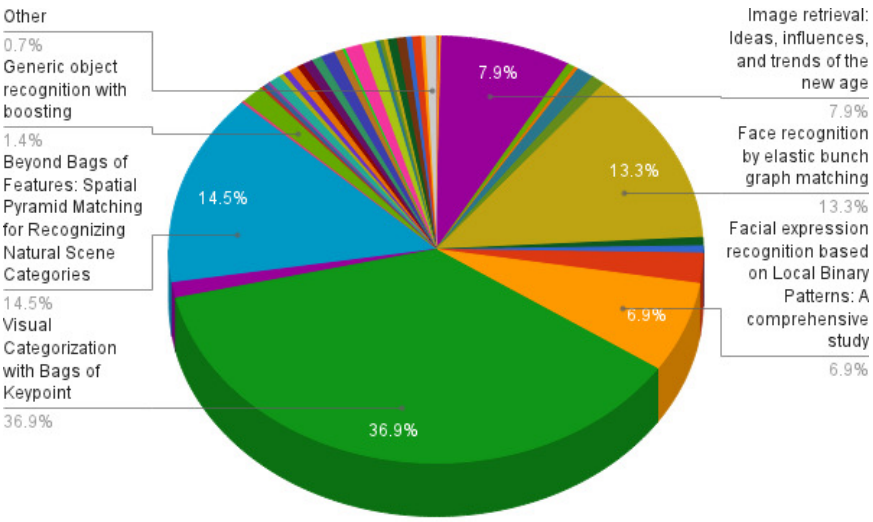

Figure 4. Papers per Amount of Citations

In [17], the authors points that the recognition of object classes is an important open problem in computer vision field. To fill this gap, they suggest tree questions that need to be answered by a classification system:

1) What points or regions should be detected?
2) Which discriminative descriptors must be extracted from these points or regions?
3) Which powerful learning algorithms are able to differentiate between the extracted descriptors?

A key point to answer these questions is the ability to describe a patch (an image region). This is crucial for many recognition algorithms because: i) it finds correspondences between different views of the same object or to represent parts of object categories and, ii) it must be robust to support changes in illumination, moderate pose variation, variation and intra-class appearance. A standard approach to this is to describe a patch using Histograms of Gradients (HoG) [18], on a combination of positions, orientations and scales. We can highlight, for example, the interest point detector / descriptor techniques named SIFT (Scale Invariant Feature Transform) [19] and GLOH (Gradient Location and Orientation Histogram) [20] which proved to be very effective for recognizing object instances [21].

To obtain information retrieval from computer images, a common requirement is to extract its features (represented by numeric values) such as shape, color, texture and local. A simple fusion of features via concatenation can result in multidimensional vector features. Currently, researchers still find it difficult to determine which is the best method of fusion of features that can produce best results. A simple concatenation can generate problems such as “curse of dimensionality”. To avoid this, the selection of discriminative features from the whole combination is essential. The challenge is to find a subset of discriminative features that provide the best recognition performance [22].

B. Methods and Techniques

Some methods used to the recognition of object classes, use histograms to evaluate the distribution of colors, gabor filters to extract shapes and wavelets transforms to improve the colour features [23]. To minimize the semantic gap, some authors proposed a comparison of object silhouette, structural feature matching, semantic level matching and learned-based approaches. These are global representations which are easy to build and are invariant to object position. However, these are very rude representation [12], [11].

The Relevance Feedback (RF) is a feature used in some information retrieval systems. The idea behind relevance feedback is to take the results that are initially returned from a given query and to use information about whether or not those results are relevant to perform a new query. We can usefully distinguish between three types of feedback: explicit feedback, implicit feedback, and blind or “pseudo” feedback. After that, the system can refine its subsequent searches [12].

Explicit feedback is obtained from assessors of relevance indicating the relevance of a document retrieved for a query. This type of feedback is defined as explicit only when the assessors (or other users of a system) know that the feedback provided is interpreted as relevance judgments. Users may indicate relevance explicitly using a binary or graded relevance system. Binary relevance feedback indicates that a document is either relevant or irrelevant for a given query. Graded relevance feedback indicates the relevance of a document to a query on a scale using numbers, letters, or descriptions (such as “not relevant”, “somewhat relevant”, “relevant”, or “very relevant”). Graded relevance may also take the form of a cardinal ordering of documents created by an assessor; that is, the assessor places documents of a result set in order of (usually descending) relevance. An example of this would be the SearchWiki [24] feature implemented by Google on their search website.

The relevance feedback information needs to be interpolated with the original query to improve retrieval performance, such as the well-known Rocchio Algorithm [25]. A performance metric which became popular around 2005 to measure the usefulness of a ranking algorithm based on the explicit relevance feedback is NDCG [26]. Other measures include precision at k and mean average precision.

Implicit feedback is inferred from user behavior, such as nothing which documents they do and do not select for viewing, the duration of time spent viewing a document, or page browsing or scrolling actions [27]. The key differences of implicit relevance feedback from that of explicit include: i) the user is not assessing relevance for the benefit of the IR system, but only satisfying their own needs and ii) the user is not necessarily informed that their behavior (selected documents) will be used as relevance feedback. An example of this is the Surf Canyon browser extension [28], which advances search results from later pages of the result set based on both user interaction (clicking an icon) and time spent viewing the page linked to in a search result.

In [11] the authors also presents a search type based both on the user as on the system point of view. The user can define its search from: keywords, image, free sketches or a combination of previous forms. The system can process the queries with: text interpretation, content-based or interactive method.

The paper [29] presents an approach for recognition in 3D that shows better performance than other approaches. A 3D SURF extension as a local descriptor technique that proved to be effective in 2D and could also be implemented in 3D. A new method for extraction of local features and descriptors for 3D shapes was also presented. The author notes that, in a video, 3D detection can be more effective with the 2D analysis of each frame.

The use of global features is typically used to recognition of three-dimensional objects classes. We can highlight some these techniques as, for example: Fourier or Spherical Harmonic, Shape Moments and Shape Histograms [29]. The problems of global representation were solved by local representation, by the means of patches. These patches are just image parts selected by interest points which detect structures such as corners, spots and peaks. Its popularity is due to the ability to adapt to scale variation. They intend to model the distribution or to characterize the distribution properties of color pacthes, gray scales and filter banks which describes local textures (using techniques such as: SIFT, texture, geometric blur and PCA-SIFT) [12].

In [30] the author contributed to the line of research that argues that a single model that combines multiple sources of information features can generate better results in the detection of object classes. The method proposed in the paper differs from others by considering global and spatial features in addition to local features. To represent local features, they used PCA-SIFT because they found that it has superior performance when compared to SIFT. The method proposed in [30] combines three popular recognition methods, texture, global shape, and PSR features with an AdaBoost model [31]. The author thinks that these techniques have complementary strengths and that should be used together. The author used the Caltech database [32] to attest the better performance of the proposed method than previous ones. The Caltech database has become a benchmark for recognition methods class. The tests also indicate that different classes require different recognition methods.

Usually, databases containing images with complex and varying backgrounds do not presented a good performance with the use of local features. Based on global features, such as shape context, some methods present problems when the object to be detected appears in a scene with cutting. The method presented by [30] will therefore tend to use global features when an object can be well defined, and to use local features when the object is cut. The author shows that the results obtained were better than those of the works which are state-of-the-art.

In [33] the authors say that the use of appropriate features is important for techniques based on SVM and KPCA in different kinds of objects [34]. For this, they used as proof of concept a robot that receives instructions to detect a specific object and to detect the class of such objects is used. In the experiment there is no method of recognizing objects that might work equally for different types of objects and backgrounds perceived by a robot. Therefore, the technique should work with multiple methods, adapting to the characteristics of objects. These objects can be classified as those that have texture, those that are texture-free and those with plain body that are recognized with SIFT. For specific objects they use KPCA + SVM. The authors also say that for recognition of object class some works uses Kernel PCA-based and SVM features. In the list of features typically used for recognition we can highlight: intensity, gabor feature and color. The objects are classified into five categories by the author and, for each category, a subset of techniques is applied to achieve the better performance.

In [16] the author proposes a hierarchical boosting framework whose experiments on a specific domain of natural images have obtained very positive results. The performance of image classifiers largely depends on two inter-related issues: (1) suitable frameworks for image content representation and automatic feature extraction; (2) effective algorithms for image classifier training and feature subset selection. To address the first issue, a multiresolution grid-based framework is proposed for image content representation and feature extraction to bypass the time-consuming and erroneous process for image segmentation. To address the second issue, a hierarchical boosting algorithm is proposed by incorporating feature hierarchy and boosting to scale up SVM image classifier training in high-dimensional feature space. The high-dimensional multi-modal heterogeneous visual features are partitioned into multiple low-dimensional single-modal homogeneous feature subsets and each of them characterizes certain visual property of images. For each homogeneous feature subset, principal component analysis (PCA) [35] is performed to exploit the feature correlations and a weak classifier is learned simultaneously. After the weak classifiers for different feature subsets and grid sizes

are available, they are combined to boost an optimal classifier for the given object class or image concept, and the most representative feature subsets and grid sizes are selected.

The segmentation consists of identifying shapes. It is a problem that can be described by the means of a graph, where the nodes are image pixels and the edges represents the relationship between a pair of pixels. Shi and Malik [36] proposed a segmentation technique cut-based that uses the contour and texture variation. In this scenario, the search for patterns in medical imagery has been being a huge research goal. The segmentation introduces some challenges, as: computational complexity, confiability on good segmentation, safe methods to assess the good segmentation. One alternative indicated by the author would be to low the segmentation dependency, including to develop techniques with no need for segmentation [11]. The paper provides an algorithm for partitioning grayscale images into disjoint regions of coherent brightness and texture. Natural images contain both textured and untextured regions, so the cues of contour and texture differences are exploited simultaneously. Contours are treated in the intervening contour framework, while texture is analyzed using textons. Each of these cues has a domain of applicability, so to facilitate cue combination we introduce a gating operator based on the texturedness of the neighborhood at a pixel. Having obtained a local measure of how likely two nearby pixels are to belong to the same region, they use the spectral graph theoretic framework of normalized cuts to find partitions of the image into regions of coherent texture and brightness.

In [11], a feature is a term that is associated with some visual property of an image, a sub-set of pixels with some semantics. There are techniques that try to extract visual signature of an image from its color. For such a method it is used color spaces more compatible with the human eye, such as LUV. Features based on textures aims to capture the granularity and repeating patterns on the surfaces of images. It is specially effective in specific domains such as: aerial images and medical images. Hence derived the field called texture features, studied within the areas of of image processing, computer vision and computer graphics.

In [37], Objects or visual classes categories are represented by a combination of local descriptors (features computed over a a limited spacial support) and their special distributions, sometimes refereed as part-based models. In computer vision, local descriptors, by being resilient to partial visibility and concealment, have proved to be well adapted to matching and recognition tasks. These tasks require descriptors that are repeatable (able to identify and detect corresponding points between two instances of an object despite changes). They motivated the development of point detectors invariant to affinity / scale and descriptors resilient to variations in illumination and geometry.

In [38], the author highlights that local descriptors with dense samples have excellent performance and therefore have become popular for object class recognition. The author indicates that, once the computer processing power increases, techniques based on sliding windows becomes more feasible for real time applications. This kind of technique is more efficient than others, despite being criticized because it uses a lot of computational resources. As proof of concept, the paper presented an implementation of Histograms of Oriented Gradients (HOG) using a computation technology called General Purpose for Graphic Processing Unit (GPGPU). HOG descriptors are features developed for object class detection that, when combined with SVM classifiers, become one of the best detectors available. The author shows that, using parallel architectures that can be found in many current graphics processors, is possible to achieve a gain of performance over than 30 times.

Other important set of techniques involving object class detection include: bag of features, constellation model, star topology, sparse texture representation and texture recognition. In [37], the author shows an approach called bag of key points for visual categorization, that is a histogram of the number of occurrences of particular images in a given image. In [39] the author investigated the quantization vector in small square. The paper propose a new framework termed Keyblock for content-based image retrieval, which is a generalization of the text-based information retrieval technology in the image domain. In this framework, methods for extracting comprehensive image features are provided, which are based on the frequency of representative blocks, termed keyblocks, of the image database. Keyblocks, which are analogous to index terms in text document retrieval, can be constructed by exploiting the vector quantization (VQ) method which has been used for image compression. By comparing the performance of the proposed approach with the existing techniques using color feature and wavelet texture feature, the experimental results demonstrate the effectiveness of the framework in image retrieval. The author found that the use of these features has a better result than approaches based on color and texture, when combined with similar vector, histogram and n-gram-models of text retrieval. The tests showed that the method is robust to background occlusion and produces good categorization even without exploring geometric information. The results with SVM are clearly superior comparing to Naive Bayes classifier [40]. As of the date of such publication, in 2004, this was the largest comparison made between these two classifiers [37].

In [41], the author proposes a technique based on bag of features, which presents a holistic approach to classify images. The method showed promising results when tested on three large sets of images. The paper presents a method for recognizing scene categories based on approximate global geometric correspondence. This technique works by partitioning the image into increasingly fine sub-regions and computing histograms of local features found inside each sub-region. The resulting “spatial pyramid” is a simple and computationally efficient extension of an orderless bag-of-features image representation, and it shows significantly improved performance on challenging scene categorization tasks. Specifically, the proposed method exceeds the state of the art on the Caltech database and achieves high accuracy on a large database

of fifteen natural scene categories. The spatial pyramid framework also offers insights into the success of several recently proposed image descriptions, including Torralba's "gist" [42] and Lowe's SIFT descriptors [17].

The technique proposed in [41] differs from multiresolution histogram because involve making a sub-sampled repeatedly in an image and compute the global histogram of the pixel values in each value. This difference gives to the proposed technique one advantage, that is not a trivial achievement. This work also highlighted the importance of discriminative scene global statistics, even in data that changes very often. The author also highlight the importance of developing methods to take advantage of this information, either with individual scene classes, or modules in conjunction with object recognition systems or with tools to evaluate the presence of potential bias in news data sets [41].

In [11], the authors proposed some techniques for the construction of features signature. In these techniques, we can highlight: region-based signature [43], histograms [44], continuous density function [45], stochastic spacial model (sophisticated, necessary in some cases, but computationally expensive) [46]. In this scenario, the multiresolution histogram has been used successfully in textured images recovery. The problem of using histograms is that they ignore where a color was found. To resolve this problem, the EMD (Earth Mover's Distance) [47] was proposed. The author highlights the region based signature as the most computationally efficient method for the recovery task. The motivation for using the region based signature is that a region more or less homogeneous (considering color and texture) generally consists of an object.

In [48], the author investigate a new method of learning part-based models for visual object recognition, from training data that only provides information about class membership (and not object location or configuration). This method learns both a model of local part appearance and a model of the spatial relations between those parts. In contrast, other work using such a weakly supervised learning paradigm has not considered the problem of simultaneously learning appearance and spatial models. Some of these methods use a "bag" model where only part appearance is considered whereas other methods learn spatial models but only given the output of a particular feature detector. Previous techniques for learning both part appearance and spatial relations have instead used a highly supervised learning process that provides substantial information about object part location. We show that our weakly supervised technique produces better results than these previous highly supervised methods. Moreover, the paper investigate the degree to which both richer spatial models and richer appearance models are helpful in improving recognition performance. The results show that while both spatial and appearance information can be useful, the effect on performance depends substantially on the particular object class and on the difficulty of the test database.

Inclusion of local color information in generic object recognition is ignored by almost all approaches, although it is important and can improve the recognition performance. In [49], the author presents a generic object recognition approach using boosting as a learning technique. Simple local color descriptors combined with the SIFT descriptors are used. Experiments using benchmark and complex generic object datasets are performed, and good performance is obtained. This is another approach to dealing with the problems of object recognition such as: variations in scale, occlusion, and appearance of objects, beyond the difficulties about the intra-class and inter-class variations. The general purpose in this paper is a type of generic object recognition using a technique such as boosting layer underlying learning. The technique consists in the detection of regions of interest from training images.

Automatic facial expression analysis is an interesting and challenging problem, and impacts important applications in many areas such as human–computer interaction and data-driven animation. Deriving an effective facial representation from original face images is a vital step for successful facial expression recognition. In [50], the authors empirically evaluate facial representation based on statistical local features, Local Binary Patterns, for person-independent facial expression recognition. Different machine learning methods are systematically examined on several databases. Extensive experiments illustrate that LBP features are effective and efficient for facial expression recognition. We further formulate Boosted-LBP to extract the most discriminant LBP features, and the best recognition performance is obtained by using Support Vector Machine classifiers with Boosted-LBP features. Moreover, the authors investigate LBP features for low-resolution facial expression recognition, which is a critical problem but seldom addressed in the existing work. We observe in our experiments that LBP features perform stably and robustly over a useful range of low resolutions of face images, and yield promising performance in compressed low-resolution video sequences captured in real-world environments.

Patch descriptors are used for a variety of tasks ranging from finding corresponding points across images, to describing object category parts. In [21], the authors propose an image patch descriptor based on edge position, orientation and local linear length. Unlike previous works using histograms of gradients, the proposed descriptor does not encode relative gradient magnitudes. The proposed approach locally normalizes the patch gradients to remove relative gradient information, followed by orientation dependent binning. Finally, the edge histogram is binarized to encode edge locations, orientations and lengths. Two additional extensions are proposed for fast PCA dimensionality reduction, and a min-hash approach for fast patch retrieval. The proposed algorithm produces state-of-the-art results on previously published object instance patch data sets, as well as a new patch data set modeling intra-category appearance variations.

Classifying the unknown image into the correct related class is the aim of the object class recognition systems. Two main points should be kept in mind to implement a class recognition system. Which descriptors that have a higher discriminative power that needs to be extracted from the images? Which

classifier can classify these descriptors successfully? The most famous image descriptor is the Scale Invariant Feature Transform (SIFT). Although, SIFT has a high performance, it is partially an illumination invariant. Adding local color information to SIFT descriptors are then suggested to increase the illumination invariant, these descriptors can be called color SIFT descriptors. In this paper, different color SIFT descriptors were implemented to evaluate their performance in the object class recognition systems. This is due to the fact that some descriptors may have a good performance in one class and bad performance in another class at the same time. All possible combinations of these descriptors were used. Some combinations of color SIFT descriptors achieved remarkable classification accuracy. Non linear x^2-kernel support vector machine is used as a learning classifier and bag-of-features representation is used to represent the image features in [17].

In [22] the authors investigates the effects of feature selection via dimensionality reduction techniques for the task of object class recognition. Two filter-based algorithms are considered namely Correlation-based Feature Selection (CFS)[51] and Principal Components Analysis (PCA). A Support Vector Machine is used to compare these two techniques against classical feature concatenation, based on the Graz02 dataset [52]. Experimental results show that the feature selection algorithms are able to retain the most relevant and discriminant features, while maintaining recognition accuracy and improving model building time.

C. Domain-Specific Papers

As shown in Figure 13, the face recognition domain was the most recurrent domain in the search. In [53], the author present a system for recognizing human faces from single images out of a large database containing one image per person. Faces are represented by labeled graphs, based on a Gabor wavelet transform. Image graphs of new faces are extracted by an elastic graph matching process and can be compared by a simple similarity function. The system differs from the preceding one [54] in three respects. Phase information is used for accurate node positioning. Object-adapted graphs are used to handle large rotations in depth. Image graph extraction is based on a novel data structure, the bunch graph, which is constructed from a small get of sample image graphs.

Viola-Jones [55] approach to object detection is by far the most widely used object detection technique because of speed of detection in images with clutter. SVM-based object detection techniques have the disadvantage of slow detection speeds because of exhaustive window search. Appearance-based detection techniques do not generalize well in the presence of pose variations. In [56], the authors propose a feature-based technique which classifies salient-points as belonging to object or background classes and performs object detection based on classified key points. Since keypoints are sparse, the technique is very fast. The use of SIFT descriptor provides invariance to scale and pose changes.

In [50], the author focuses on the problem of automatic facial expression analysis. It was pointed that extracting an effective facial representation from images of real faces is a vital step for the successful recognition of facial expressions. Two common approaches are those based in geometric features and appearance-based methods. The author in this paper empirically studied the facial representation based on features named Local Binary Pattern (LBP) for facial expression recognition independent from person.

For this purpose the markov random field [57] was used. The paper consider a texture to be a stochastic, possibly periodic, two-dimensional image field. A texture model is a mathematical procedure capable of producing and describing a textured image. The paper explore the use of Markov random fields as texture models. The binomial model, where each point in the texture has a binomial distribution with parameter controlled by its neighbors and "number of tries" equal to the number of gray levels, was taken to be the basic model for the analysis. A method of generating samples from the binomial model is given, followed by a theoretical and practical analysis of the method's convergence. Examples show how the parameters of the Markov random field control the strength and direction of the clustering in the image. The power of the binomial model to produce blurry, sharp, line-like, and blob-like textures is demonstrated. Natural texture samples were digitized and their parameters were estimated under the Markov random field model. A hypothesis test was used for an objective assessment of goodness-of-fit under the Markov random field model. Overall, microtextures fit the model well. The estimated parameters of the natural textures were used as input to the generation procedure. The synthetic microtextures closely resembled their real counterparts, while the regular and inhomogeneous textures did not.

The author in [48] also shows that models based on bags performs better than spatial models for the most common data sets, but the results here suggest that this is due to characteristics of the datasets. But those differences must to be more studied to be more well characterized, in order to determine which aspects of bag models in comparison to spatial models are responsible for these differences.

IV. Correlated Works

In [58], Nascimento has proposed a system for automatic detection of objects in images, in order to indicate the presence or absence of a particular object. The author tried to cover the difficulties involved in the detection process of objects, such as object rotation and translation, scale, and lighting. To do this, the author conducted an analysis of the available techniques then implemented algorithms known and finally evaluated results obtained.

The author has exposed different techniques available for the object detection task, and generated as a result a guide to help identify the technique that better fits some domain. His work differs from that proposed in this project since it is an evaluation of existing techniques, and the project proposed here consists of an integrated tool to be use by developers and/or end users.

In [**?**], Maia presents some methods based on Local Descriptors using a representation of vision systems that do not restrict the objects to be detected/recognized by its pose and size. In this scenario, he conducts an analysis through a case study using the proposed methodology.

The Maya's work differs from the work proposed here by not addressing the issue of the specifics issues that each class of objects can present. In addition, a developer would have to be trained in order to understand the technical proposal, and then implement the final solution.

In [59], Felzenszwalb presents a cutting edge system to find objects in cluttered images, introducing the use of latent variables that allow objects to be recognized even with variations in their appearance. This increases the detection efficiency but complicates the training task.

The Felzenszwalb's work differs from the work proposed here by not presenting a solution that simplifies the developer's task, even consisting of a solution that improves the performance of object detection for certain classes, it not necessarily wants to cover all of them. The library proposed here aims to address the specificities of each object class through the use of domain specific detectors.

In [60], the Bertrand presents the PyCVF. It is an open source framework that provides basic functionality for computer vision and video mining, such as: application processing, indexing multimedia data sets, models of training and search for results in a model.

The problem with this work within the context of this paper is that it requires prior knowledge from developers about the functioning of object recognition techniques, such as training, and the use of models. The integrator proposed here aims to provide a transparent recognition of objects for the developer and end user. PyCVF or another library like OpenCV[61] could be used by our detectors.

In [16], Gao aims to provide a suitable framework for representing content and automatic feature extraction. For this he uses a multi-resolution framework based on grid for image content representation and feature extraction. This in combination with a hierarchical boosting algorithm using the SVM classifier in a multi-dimensional feature space. These features are subdivided into multiple spaces with few dimensions on which they apply the Principal Component Analysis (PCA), in order to explore correlations between features, and train a weak classifier simultaneously. The author performed experiments within the realm of natural images, and claims to have obtained positive results, and that the procedure can be successfully applied to other image areas.

In [62], Leibe presents a model for detecting multiple object classes through the use of Generative Models. The author claims to have contributed to an efficient object recognition system capable of recognizing and simultaneous locating of several object classes.

The solutions proposed by [16] and [62] have focused the techniques details involved in the detection of object classes task. They actually do an important contribution as they achieved the state of the art in the application of the available techniques. Although they have delivered the developer a way forward to make the object detection in a domain, their solutions can not be transparent about the techniques details like training, use of classifiers and automatic annotation. And then, requires prior knowledge on Computer Vision aspects.

The solution proposed here also intends to allow detection of multiple classes of objects through the use of various detectors, but one for each domain, allowing the user to combine them for better results.

V. Integrator Tool Description

The solution proposed here is based on a software integrator arranged in three layers, as shown in Figure 6. The main layer contains the integrator, whose function is to receive commands through parameters and redirect them to the detectorś layer. On this layer are the detectors, each detector is a piece of software programmed to recognize objects within a certain scope. To exemplify, in Figure 6, there exists one detector for each of the following domains: vegetation, people and vehicles. Each detector in turn, can use thirdy-party libraries (like OpenCV) from the lib's layer to provide the required functionality.

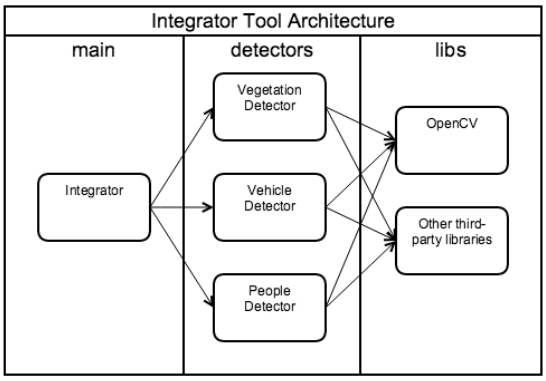

Figure 6. Integrator Architecture

The integrator consists of an executable program which receives a path to an image and some configuration commands as input parameters. The detectors folder indicates detectors known, as shown in Figure 7. The commands expected by the integrator aims at selecting which detectors will be used for the detection task and to set the output format generated.

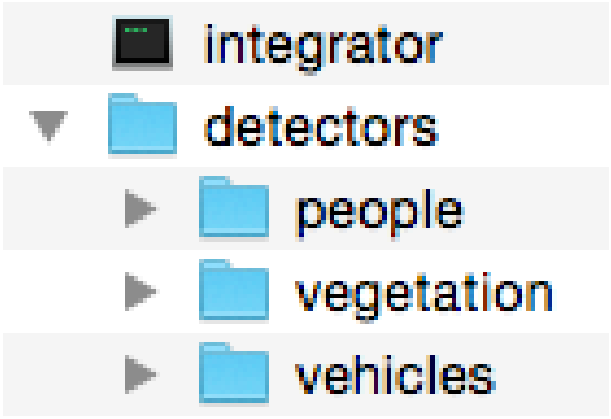

Figure 7. Integrator Folder Structure

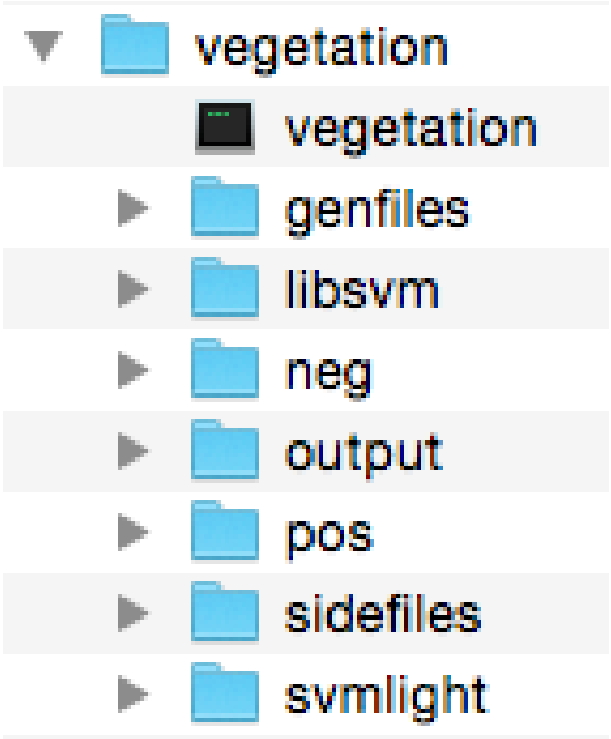

Figure 9. Estrutura do Detector de Vegetação

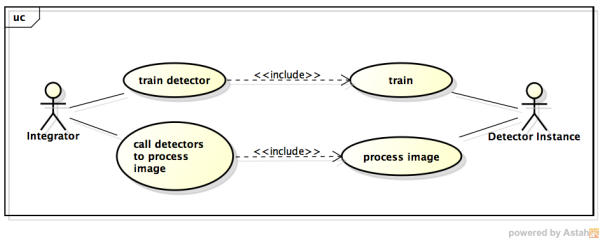

Figure 8. Integrator Use Case

Each detector is a self contained software, specialized for a domain, previously trained, and has varying complexity. Its purpose is to carry out specific detection of a domain object, as shown in Figure 9, you can develop a detector to know how to identify vegetation from a top view. This detector may detail, for example, if the object encountered is a shrub, tree or lawn. Each directory contains a detector folders with images used in training (both positive and negative examples), and files generated by the training stage, for use in the detection step. As can be seen by Figure 8, the plugin must provide a function for your classifier can be retrained, and a function to process an image. Any other dependencies that the detector has to be placed in this directory.

Here it is important to emphasize the importance of the independence of each detector, they can encapsulate details of varying complexities, and use the techniques, libraries and algorithms that best apply to the treated area. After processing an image, each detector will return the objects found in the format specified in Annex 1, plus any other relevant information.

After each detector have returned the objects found, the integrator will concatenate the responses received and generate a final response to the initial request, as can be seen in the activity diagram Figure 10. In this way, the programmer can use integrator to perform object detection ignoring the complexity of the detectors involved in the process. He needs just to worry about dealing with the information returned, which indicates each object found, its class and its position within the image.

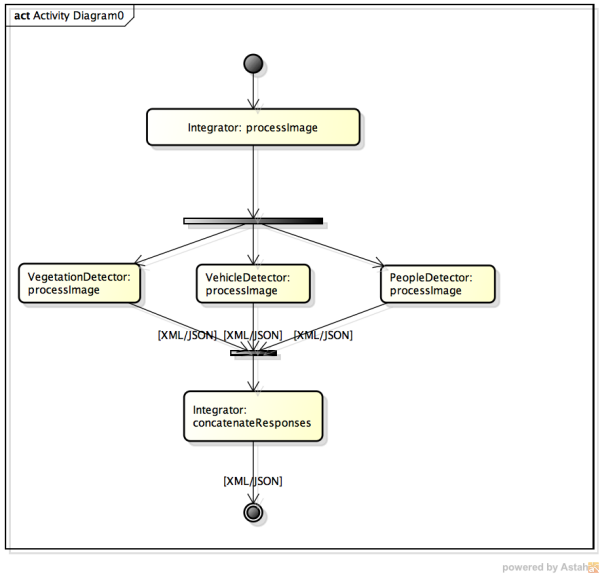

Figure 10. Diagrama de Atividade - processarImagem()

Each detector should provide a set of object types that it can detect, and for each object a set of attributes in xml or json format, see Annex 1.

Each detector should consist of an executable file of a particular platform (eg. windows, mac or linux), the example in Figure 9, the executable file is textit vegetation. To run it should make the 'vegetation' command using the following options:

- detect [attributes]: the return of this command should be some text(xml or json) containing the detected objects. Possible attributes:
  - -image: specifies an image to detect objects.
  - -video: specifies a video for object detection. Each tag object in this case is increased by the attribute "time".

The idea is that each executable is autonomous and perform the task of detection using a classifier already trained and suitable for the detection. The user of the integrator tool described here will tell which detectors should be used specifying a command as described in Annex 2.

The command will return a xml or json file containing a set of elements as defined in Annex 1. With these the developer of user information may make subsequent computations you want.

Through the class diagram Figure 11, it is possible to check the classes implemented in this solution. The class *IntegratorTool* by means of the method *readDetctors* query the folder *detectors*, which contains all known detectors. Each detector is called through a system call, so each one must implement the *DetectorInterface* interface, which defines the methods *train()* and *processImage()*, and provides a concrete method *main()*, who handles the parameters and correctly call the mentioned methods. To develop a new detector named *people_detector* the developer must follow the steps:

1) Create a *PeopleDetector* class (similar to the class *VehicleDetector* on Figure 11);
2) Implement the *DetectorInterface* interface and generate an executable to forward the parameters received for the method *main*, defined within the interface;
3) Finally, the generated executable must be named *people_detector* and positioned in a directory named *people_detector*, inside the *detectors* directory, as shown in Figure 7.

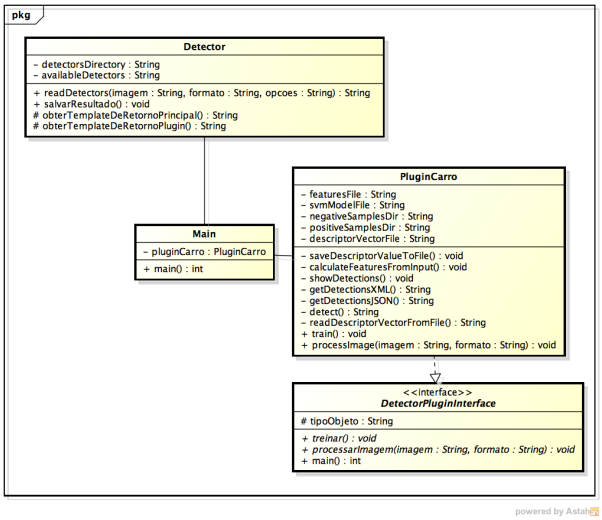

Figure 11. Diagrama de Classes

Below are three examples to illustrate the use of the integration tool. For these examples it is assumed that the following detectors are available: vehicle detector, detector vegetation, people detector, see Figure 7. All input images are taken from a top view. Examples:

- In this example, there is a picture of a forest area photographed from a helicopter and you want to detect the amount of trees contained in that image to determine the level of deforestation, see Figure 12.

Figure 12. Input picture for Vegetation Detector

To this end, one can use the integrator tool informing the use of the vegetation detector:

1) detect -enable=vegetation -image=forest_picture.png –format=json

- In this example, there is an image captured from a crossing between avenues from the top of a building and one wishes to count the number of vehicles and pedestrians in each crossing side for generating traffic information, see Figure 13.

Figure 13. Input picture for Vehicles and Person detectors. Source: www.tribunadonorte.com.br

To this end, one can use the tool integrator informing the use of both vehicle and people detectors. This command can be passed in two ways: enabling two detectors directly or disabling the vegetation detector:

1) detect -enable=vehicle,people -image=crossing_picture.png -format=json
2) detect -disable=vegetation -image=crossing_picture.png -format=json

- The third example, also uses a crossing in a urban scene, but includes the need to extend the detection of objects in order to evaluate the amount of vegetation of a city, besides generates traffic information, see Figure 14.

Figure 14. Input picture for all detectors. Source: www.atibaianovo.com.br

In this case, all available detectors should be used, the command will be:

1) detect -enable-all -image=crossing_picture2.png -format=json

For any of the above cases, if the developer decides to research about the techniques, algorithms and libraries available for object detection, he would have to answer questions like: How do you characterize the human vision algorithmically? What are the available libraries that best fit the problem? How to train a classifier to recognize a certain class of objects? How to prepare images for training? How to optimize the selection of images to get better? Which features to seek? Which detector to use? What classifier to use? One should use segmentation? The final algorithm has satisfactory computational performance? How to treat the semantic gap and the sensorial gap? [11] [37]

Each object class domain may answer to the questions above in a particular way. A systematic survey of the literature on the subject exhibited a wide range of techniques and cases where object detection has been successfully applied, and showed how to adjust the parameters of the algorithms for the dealt domain. However, this is not a trivial task, and depending on the issues presented by the image database in hand, some effort to reach the ideal solution may be necessary.

The solution proposed here states that experienced developers in detection of object classes within a certain area should develop a detector that encapsulates all the complexity and know-how involved in this task. Thus, new developers would use this detector, in order to avoid the expenditure of time and energy on expertise in matters that are unfamiliar to them and perhaps, even with dedication, they probably would failure to reach a satisfactory result due to incompatible professional profile, lack of competence or shortly dedication.

VI. Conclusion and Future Works

This work aimed to propose an integrative tool for recognition of object classes in digital images. As a result introduced a systematic mapping of the area where the most referenced work area were found, and the most used techniques were classified according to their functionality and performance. It was found that the recovery of semantic information from images is not a trivial task. This task involves challenging issues such as: semantic gap, sensorial gap and computational cost. The object class detection in images still can bring other challenges, such as variations intra-class and inter-class. Several techniques of feature extraction and machine learning have been developed to achieve satisfactory results.

The integrator tool delivered a very simple interface, useful to developers and to end-users. In addition, to develop a satisfactory solution for a given class of objects, it was shown that a developer needs to have knowledge about algorithms and recognition techniques best suited to the desired object classes. What is a considerable difficulty for developers who have no formal training in computer vision.

The solution proposed here achieves its goals, while describing an environment that offers recognition of ready-made objects class solutions for specific areas, made by experts, and suitable for use by non-expert developers.

For future work, in order to further contribute to the field of ubiquitous systems, a extension library would be made to provide a technical environment for implementing recognition of object classes adapted for the limited resources on mobile devices. The idea is to avoid data transferring over Internet to a remote server, and use some *ad hoc* network communication between devices, as proposed by [4].

Another possible future work is to identify the main demands recognition of objects of the most common classes in ubiquitous applications, and to develop a set of plugins for these classes in order to make a this integrator tool a useful piece of software from a practical point of view.

VII. Annex

Annex 1. Detector's output formats

XML:

```
<plugin name="detector_vegetacao">
<objects>
<object type="arvore">
<coords x="99" y="99" w="64" h="128"/>
</object>
<object type="arbusto">
<coords x="99" y="99" w="64" h="128"/>
</object>
<object type="gramado">
<coords x="99" y="99" w="64" h="128"/>
</object>
</objects>
</plugin>
```

JSON:

```
[
"plugin_name" : "detector_vegetacao",
"objects" : [
"type" : "arvore",
"coords" :  "x" : 99, "y": 99, "w":64, "h" : 128
,
"type" : "arbusto",
"coords" :  "x" : 99, "y": 99, "w":64, "h" : 128
,
"type" : "gramado",
"coords" :  "x" : 99, "y": 99, "w":64, "h" : 128
]
]
```

Annex 2. A utilização da biblioteca de plugins por meio do desenvolvedor se dará através do uso do comando 'detector_principal', com as opções abaixo:

- detect [atributos]: o retorno desse comando deverá ser o xml ou json contendo os objetos detectados. Atributos possíveis:
  1) –enable-all = habilita todos os plugins
  2) –disable-all = desabilita todos os plugins
  3) –enable=X,Y = habilita os plugins X e Y
  4) –disable=Z,W = desabilita os plugins Z e W
  5) –image: especifica uma imagem para se detectar objetos.
  6) –video: especifica uma vídeos para detecção de objetos. Cada tag objeto neste caso vem acrescida do atributo "time".
  7) –format: especifica o formato de retorno dos objetos detectados: xml ou json.

References

[1] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.

[2] X. Zhang, Y.-H. Yang, Z. Han, H. Wang, and C. Gao, "Object class detection: A survey," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 10:1–10:53, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2522968.2522978

[3] J. G. R. Maia, "Detecção e reconhecimento de objetos usando descritores locais," PhD in Bioinformatics, Universidade Federal do Ceara, Biomathematics Group, R. Qta. Grande 6, 2780-156 Oeiras, Portugal, 2010, iSBN 9729961506.

[4] H. Wirtz, J. Rüth, and K. Wehrle, "Facilitating direct and ubiquitous mobile computer vision," in *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '14. New York, NY, USA: ACM, 2014, pp. 199–207. [Online]. Available: http://doi.acm.org/10.1145/2677972.2677974

[5] G. R. Langley, "Evidence-based medicine: How to practice and teach ebm," *CMAJ: Canadian Medical Association Journal*, vol. 157, no. 6, p. 788, 1997.

[6] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - a systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.infsof.2008.09.009

[7] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and software technology*, vol. 50, no. 9, pp. 833–859, 2008.

[8] M. N. Islam, "A systematic literature review of semiotics perception in user interfaces," *Journal of Systems and Information Technology*, vol. 15, no. 1, pp. 45–77, 2013.

[9] A. Fernandez, E. Insfran, and S. Abrahão, "Usability evaluation methods for the web: A systematic mapping study," *Information and Software Technology*, vol. 53, no. 8, pp. 789–817, 2011.

[10] H. M. Cooper, "Organizing knowledge syntheses: A taxonomy of literature reviews," *Knowledge in Society*, vol. 1, no. 1, pp. 104–126, 1988.

[11] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Comput. Surv.*, vol. 40, no. 2, pp. 5:1–5:60, May 2008. [Online]. Available: http://doi.acm.org/10.1145/1348246.1348248

[12] J. Krapac and F. Jurie, "Learning distance functions for automatic annotation of images." in *Adaptive Multimedia Retrieval*, ser. Lecture Notes in Computer Science, N. Boujemaa, M. Detyniecki, and A. Nürnberger, Eds., vol. 4918. Springer, 2007, pp. 1–16. [Online]. Available: http://dblp.uni-trier.de/db/conf/amr/amr2007.html

[13] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic *et al.*, "Query by image and video content: The qbic system," *Computer*, vol. 28, no. 9, pp. 23–32, 1995.

[14] T. Gevers and A. W. Smeulders, "Pictoseek: Combining color and shape invariant features for image retrieval," *Image Processing, IEEE Transactions on*, vol. 9, no. 1, pp. 102–119, 2000.

[15] J. R. Smith and S.-F. Chang, "Visualseek: a fully automated content-based image query system," in *Proceedings of the fourth ACM international conference on Multimedia*. ACM, 1997, pp. 87–98.

[16] Y. Gao, J. Fan, X. Xue, and R. Jain, "Automatic image annotation by incorporating feature hierarchy and boosting to scale up svm classifiers," in *Proceedings of the 14th Annual ACM International Conference on Multimedia*, ser. MULTIMEDIA '06. New York, NY, USA: ACM, 2006, pp. 901–910. [Online]. Available: http://doi.acm.org/10.1145/1180639.1180840

[17] T. H. Rassem and B. E. Khoo, "Object class recognition using combination of color sift descriptors," in *Imaging Systems and Techniques (IST), 2011 IEEE International Conference on*. IEEE, 2011, pp. 290–295.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[19] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[20] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.

[21] C. L. Zitnick, "Binary coherent edge descriptors," in *Proceedings of the 11th European Conference on Computer Vision: Part II*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 170–182. [Online]. Available: http://dl.acm.org/citation.cfm?id=1888028.1888042

[22] N. Manshor, A. Halin, M. Rajeswari, and D. Ramachandram, "Feature selection via dimensionality reduction for object class recognition," in *Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), 2011 2nd International Conference on*. IEEE, 2011, pp. 223–227.

[23] W.-Y. Ma and B. Manjunath, "A comparison of wavelet transform features for texture image annotation," in *Image Processing, International Conference on*, vol. 2. IEEE Computer Society, 1995, pp. 2256–2256.

[24] P. Haase, D. Herzig, M. Musen, and T. Tran, "Semantic wiki search," in *The Semantic Web: Research and Applications*. Springer, 2009, pp. 445–460.

[25] C. Jordan and C. Watters, "Extending the rocchio relevance feedback algorithm to provide contextual retrieval," in *Advances in Web Intelligence*. Springer, 2004, pp. 135–144.

[26] E. Yilmaz, E. Kanoulas, and J. A. Aslam, "A simple and efficient sampling method for estimating ap and ndcg," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008, pp. 603–610.

[27] D. Kelly and N. J. Belkin, "Reading time, scrolling and interaction: exploring implicit sources of user preferences for relevance feedback," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2001, pp. 408–409.

[28] D. Hardtke, M. Wertheim, and M. Cramer, "Demonstration of improved search result relevancy using real-time implicit relevance feedback," *Understanding the User-Logging and Interpreting User Interactions in Information Search and Retrieval (UIIR-2009)*, p. 1, 2009.

[29] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. J. V. Gool, "Hough transform and 3d surf for robust three dimensional classification." in *ECCV (6)*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6316. Springer, 2010, pp. 589–602. [Online]. Available: http://dblp.uni-trier.de/db/conf/eccv/eccv2010-6.html

[30] W. Zhang, B. Yu, G. Zelinsky, and D. Samaras, "Object class recognition using multiple layer boosting with heterogeneous features," *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Volume 2-Volume 02*, pp. 323–330, 2005.

[31] P. Viola and M. Jones, "Fast and robust classification using asymmetric adaboost and a detector cascade," *Proc. of NIPS01*, 2001.

[32] W. Zhang, B. Yu, G. J. Zelinsky, and D. Samaras, "Object class recognition using multiple layer boosting with heterogeneous features," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 323–330.

[33] A. Mansur, M. A. Hossain, and Y. Kuno, "Integration of multiple methods for class and specific object recognition," in *Proceedings of the Second International Conference on Advances in Visual Computing - Volume Part I*, ser. ISVC'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 841–849.

[34] L. Cao, K. Chua, W. Chong, H. Lee, and Q. Gu, "A comparison of pca, kpca and ica for dimensionality reduction in support vector machine," *Neurocomputing*, vol. 55, no. 1, pp. 321–336, 2003.

[35] L. I. Smith, "A tutorial on principal components analysis," *Cornell University, USA*, vol. 51, p. 52, 2002.

[36] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *Int. J. Comput. Vision*, vol. 43, no. 1, pp. 7–27, Jun. 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1011174803800

[37] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[38] C. Wojek, G. Dorkó, A. Schulz, and B. Schiele, "Sliding-windows for rapid object class localization: A parallel technique," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, G. Rigoll, Ed. Springer Berlin Heidelberg, 2008, vol. 5096, pp. 71–81.

[39] L. Zhu, A. Zhang, A. Rao, and R. Srihari, "Keyblock: An approach for content-based image retrieval," in *Proceedings of the eighth ACM international conference on Multimedia*. ACM, 2000, pp. 157–166.

[40] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid." in *KDD*, 1996, pp. 202–207.

[41] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2169–2178. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2006.68

[42] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[43] G. Lu and A. Sajjanhar, "Region-based shape representation and similarity measure suitable for content-based image retrieval," *Multimedia Systems*, vol. 7, no. 2, pp. 165–174, 1999.

[44] G. Pass and R. Zabih, "Comparing images using joint histograms," *Multimedia systems*, vol. 7, no. 3, pp. 234–240, 1999.

[45] P. Mansfield, "Multi-planar image formation using nmr spin echoes," *Journal of Physics C: Solid State Physics*, vol. 10, no. 3, p. L55, 1977.

[46] R. Durrett, "Stochastic spatial models," *SIAM review*, vol. 41, no. 4, pp. 677–718, 1999.

[47] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[48] D. J. Crandall and D. P. Huttenlocher, "Weakly supervised learning of part-based spatial models for visual object recognition," in *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ser. ECCV'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 16–29. [Online]. Available: http://dx.doi.org/10.1007/11744023-2

[49] J. D. D. Hegazy, "Boosting colored local features for generic object recognition." SP MAIK Nauka/Interperiodica, 2008, pp. 323–327.

[50] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," *Image Vision Comput.*, vol. 27, no. 6, pp. 803–816, May 2009. [Online]. Available: http://dx.doi.org/10.1016/j.imavis.2008.08.005

[51] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, 1999.

[52] F. Moosmann, B. Triggs, F. Jurie *et al.*, "Fast discriminative visual codebooks using randomized clustering forests," *Advances in Neural Information Processing Systems 19*, pp. 985–992, 2007.

[53] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 775–779, Jul. 1997. [Online]. Available: http://dx.doi.org/10.1109/34.598235

[54] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *Computers, IEEE Transactions on*, vol. 42, no. 3, pp. 300–311, 1993.

[55] M. Jones and P. Viola, "Fast multi-view face detection," *Mitsubishi Electric Research Lab TR-20003-96*, vol. 3, p. 14, 2003.

[56] S. Shah, S. H. Srinivasan, and S. Sanyal, "Fast object detection using local feature-based svms," in *Proceedings of the 8th International Workshop on Multimedia Data Mining: (Associated with the ACM SIGKDD 2007)*, ser. MDM '07. New York, NY, USA: ACM, 2007, pp. 7:1–7:5. [Online]. Available: http://doi.acm.org/10.1145/1341920.1341927

[57] G. R. Cross and A. K. Jain, "Markov random field texture models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 1, pp. 25–39, 1983.

[58] M. C. Nascimento, "Detecção de objetos em imagens. trabalho de conclusão de curso," Universidade Federal de Pernambuco (Graduação em Ciência da Computação), 2007.

[59] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Visual object detection with deformable part models," *Commun. ACM*, vol. 56, no. 9, pp. 97–105, Sep. 2013. [Online]. Available: http://doi.acm.org/10.1145/2494532

[60] S. I. S. Bertrand Nouvel, "The python computer vision framework," http://pycvf.sourceforge.net/.
[61] "Opencv," http://opencv.org/about.html.
[62] *Multiple Object Class Detection with a Generative Model*, vol. 1, 2006. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2006.202