Integrating Ontologies and Computer Vision for Classification of Objects in Images

Daniele Porello[†], Marco Cristani[⊕], and Roberta Ferrario[†]

[⊕] Department of Computer Science, University of Verona, Italy
[†] Institute of Cognitive Sciences and Technologies of the CNR, Trento, Italy

Abstract. In this paper, we propose an integrated system that interfaces computer vision algorithms for the recognition of simple objects with an ontology that handles the recognition of complex objects by means of reasoning. We develop our theory within a foundational ontology and we present a formalization of the process of conferring meaning to images.

Keywords: Computer vision, ontology, classification, semantic gap.

1 Introduction

In general terms, we could see classification as the process of categorizing what one sees. This involves the capabilities of recognizing something that has already been seen, singling out similarities and differences with other things, and a certain amount of understanding.

As human beings, of course we learn to recognize and classify things by being exposed to positive and negative examples of attribution of instances to a class, like when we say to children "this is a cat", "this is not a cat". But, as we grow, we progressively integrate this acquired capability with a high level knowledge of which are the characteristics that can help us to classify something that we see in the right category. If the task we are involved in is that of a classification based only on visual properties, in the previous example this amounts to leveraging on descriptions like "a cat is a furry, for-legged thing, which can be colored in a restricted number of ways that include black, white and beige, among others, but not blue or green". So, if we have seen many cats in our life, probably we would not need the description and we would just use our basic capability of recognizing similar things, but if we haven't seen any cat, but we know what it means to be furry, what are legs and how such colors look like, we would probably use the description to classify something as a cat or not.

Turning now to artificial agents, we believe that, in order for them to perform in an optimal way the classification task, both these capabilities, basic recognition by repeated exposition and high level classification by following a definition, should be provided and moreover integrated, analogously as it happens for human beings.

In this paper we try to present an approach meant to endow artificial agents with these integrated capabilities for classification: we show how some things in

an image can be classified with basic concepts just by running computer vision algorithms that are able to directly recognize them, whereas other things can be classified by means of definitions in a visual ontology that aggregate the basic categories singled out by algorithms. It is noteworthy that the concepts we use to classify things based on vision are a subclass of "ordinary" concepts, as they depend on specific factors, which for humans are the visual apparatus of the subject who is seeing the things to classify, his/her familiarity with things that are similar to it, his/her background knowledge, the perspective and the conditions of sight, that may vary through time. Analogously, for artificial agents classification is influenced by the characteristics of the camera that is recording the scene, from the perspective of the camera, from the training set of the classifier (that is the counterpart of the previous exposition to similar things) and from the visual theory that provides background knowledge for classification. This means that classification through vision is a peculiar kind of classification, that gives as an output claims as "this thing *looks* like a cat" rather than "this thing *is* a cat" and this also means that different agents, being they humans or artificial, may view and then classify things with different concepts and classification may vary through time. That is, classification by means of vision is an example of "*looks*-talk", in Sellars' words [10]. It is important to keep visual concepts distinct from "ordinary" concepts, in order to be able to connect what agents know about a thing and what they know about how it looks like. This is particularly helpful when the direct visual classification is uncertain, for instance when only some parts of the thing are visible and one can deduce the presence of other invisible parts moving from the background knowledge. Moreover, when the direct classification is in disagreement with the background knowledge, the latter can drive the process of inspecting further options. In the case of artificial agents, this translates into using inferences on the visual ontology to drive the choice of the computer vision classifiers to be applied.

In the framework that we are presenting, we provide artificial agents with computer vision classifiers and with an ontology for visual classification. Roughly speaking, the computer vision classifiers will be tailored to the *basic* concepts of the ontology, which will be constituted by axioms connecting such basic concepts to form other, more complicated, *defined* concepts. The visual ontology should define how the entities classified by visual concepts look like. It is important that such visual ontology is built on the basis of a solidly grounded *foundational* ontology. This is for several reasons: first of all, this enhances interoperability, as the foundational ontology makes explicit the hidden assumptions behind the modeling; moreover, on the same foundational ontology one can build a domain ontology that expresses properties of the concepts of the domain that do not depend on the visual dimension: this allows for integrating how objects are supposed to *be* and how objects are supposed to *appear* to the relevant agent. The integration of the two is exactly what is needed to solve cases of uncertainty and disagreement mentioned earlier.

The idea to use ontologies for image interpretation is not new. Among the first efforts in this direction there are [12], [11], and [4], while more recent con-

tributions are [9] and [2]. The significant difference of our approach is that we build our treatment on a foundational ontology in order to explain the interface of computer vision techniques with ontological reasoning. In particular, we focus on the process of conferring content to an image and we show that it is a heterogeneous process that involves perception and inference.

The paper is structured as follows. In Section 2, we discuss the methodology based on foundational ontologies and we introduce the basic concepts of the ontology that we use. In Section 3, we present our modelling of the process of conferring contents to images. We do so by introducing the notion of visual theory that is the formal background that is required to ascribe meanings to images. In Section 4, we instantiate our approach by means of a toy example of ontology for talking about geometric figures. Section 5 concludes and points at possible future work.

2 An ontology for visual classification

Similarly as for humans, for the task of classification, i.e. to decide to which class something that is observed/perceived belongs to, it could be very helpful also for artificial agents to be endowed with the capability of reasoning over information coming from their visual system. This means being able to integrate different types of information: that coming from the visual system with the background knowledge. In order to do this, we propose to build a visual ontology to be integrated with a domain specific ontology, so that agents can classify entities (for instance objects) not only by directly applying a computer vision classifier for every entity that is represented in an image, but also by inferring the presence of such entity by reasoning over ontological background knowledge. For instance, the framework could allow to exclude the outcome of a visual classification if such outcome contradicts the background information by identifying an object displaying some properties that cannot be ascribed to it according to the background ontology (like identifying as a building an object that flies).

The role of a visual ontology should be that of providing a language to interface information coming from computer vision with conceptual information concerning a domain, for instance as provided by experts. How the expert's knowledge has to be collected is a rather different problem that we shall not approach here (see [9]).

One of the points of using ontologies is that of enabling the integration of different sources of knowledge. For this purpose, in the following, we shall approach a visual ontology to be used for the classification of entities in images; this will formalize the process of associating meaning to images or parts of images[1]. Once meaning is provided to images, we can use conceptual knowledge in order to reason about the content of an image, make inferences, and possibly revise the classification once more information has been provided. As a matter of fact, visual concepts share with *social* concepts the temporary nature (something is

[1] In this paper we focus only on images as a starting point, but the approach is in principle applicable to videos as well.

classified as x at time t) [8], but, differently from social concepts, they do not need an agreement by a community to be applied, as they depend primarily from the visual system (classifier). When a visual concept is attributed to a certain entity, we should interpret this attribution as "The entity x *looks as* a y at t". This also means that the visual classification may be revised through time and through the application of different classifiers.

The fundamental principles of our modeling are the following: 1. Images are physical objects; 2. Image understanding is the process of conferring meaning to images; 3. Meaning is conferred to (a part of) an image by classifying it by means of some concept.

Images are physical objects in a broad sense that includes for instance digital images. This could be seen as a controversial point, but our choice to consider them as physical objects is driven by the fact that we want to talk about physical properties that can be attributed to images or their parts, like color, shape etc. We are aware of the fact that images are processed at different levels during a classification task performed with computer vision techniques and that physical properties cannot be directly attributed at the intermediate levels of processing, but we leave the treatment of such issues for future work.

An image has *per se* no meaning, that is, no semantic content. We view the ascription of meaning as an action performed by an (artificial) agent who is classifying the image according to some relevant categories. This act of classification of an image is what we are interested in capturing by formalizing. In order to do that, we shall introduce some basic elements of the foundational ontology DOLCE [7], which provide a rich theory of concepts and of the act of classification. DOLCE is a foundational ontology and the choice of leveraging on it is also due to the fact that, given the generality of its classes, it is maximally interoperable, so applicable to different domains once its categories are specialized and tailored to such domains. Moreover, differently from most of the other foundational ontologies, it does not rely on strongly realistic assumptions. On the contrary, the aim of DOLCE is that of capturing the perspective of a cognitive agent and is thus, in our opinion, more naturally adaptable to represent the "*looks*-talk" of a visual ontology.

2.1 The top level reference ontology: DOLCE

We start by recalling the basic primitives of the foundational ontology DOLCE [7]. The reason why we focus on DOLCE is that it is a quite complex ontology that is capable of interfacing heterogeneous types of knowledge. In particular, the theory of concepts that is included in DOLCE is fundamental for our approach. We focus on the DOLCE-CORE, the ground ontology, [1]. The ontology partitions the objects of discourse, labelled *particulars* PT, into the following six basic categories: *objects* O, *events* E, *individual qualities* Q, *regions* R, *concepts* C, and *arbitrary sums* AS. The six categories are to be considered as rigid, i.e. a particular does not change category through time. For example, an object cannot become at a certain point an event. Objects represent particulars that are mainly located in space, as for instance this table, that chair, this picture of a chair. An

individual quality is an entity that we can perceive and measure that inheres to a particular (e.g. the color, the temperature, the length of a particular object). The relationship between the individual quality and its (unique) bearer is the *inherence*: $\mathrm{I}(x, y)$ "the individual quality x inheres to the entity y". The category Q is partitioned into several *quality kinds* Q_i, for example, color, weight, temperature, the number of which may depend on the domain of application. Each individual quality is associated to (one or more) *quality space* $S_{i,j}$ that provides a measure for the given quality[2]. Quality kinds can also be multi-dimensional, i.e. they can be composed by other, more specific quality kinds: e.g. the color of an object may be associated to color quality kinds with their relevant spaces, such as hue, saturation, brightness. The category of regions R includes subcategories for spatial locations and a single region for time. As already anticipated, DOLCE includes the category of concepts, which is crucial here. Concepts are in DOLCE reifications of properties: this allows for viewing concepts as entities of the domain and to specify their attributes [8]. In particular, concepts are used when the intensional aspects of a predication are salient for the modeling purposes, when for instance we are interested in predicating about the properties of a certain entity that this acquires in virtue of the fact of being classified with a certain concept. The relationship between a concept and the object that instantiates it is called *classification* in DOLCE: $\mathtt{CF}(x, y, t)$ "x is classified by y at time t". In what follows, we view qualities as concepts that classify particulars (e.g. being red, being colored, being round), thus as qualities that may be applied to different objects.

In DOLCE-CORE, we can understand predication in three senses: as *extensional classes*, by means of properties, as *tropes*, by means of individual qualities, or as *intensional classifications* by means of concepts. We shall deploy concepts in order to formalize the relationship between an image and its content. The choice is motivated by the intuition that the content of images is dependent much more on its relation with intensional aspects of the classification, like the classifier used to ascribe such content, than on its mere extensional instances. As already anticipated, we assume that images are physical objects, that is, we view an image as its mere physical substratum. The reason is that here we are interested in classifying physical qualities, such as color, shape, dimension and we want to interpret the act of conferring these qualities to an image as an act of classification of the image under these concepts.

3 Conferring content to images

In order to integrate the information coming from computer vision with information expressed in symbolic (or logical) terms, we approach the problem of conferring a meaning to an image. This problem is also known as the *semantic gap* problem in the computer vision literature [13]. We aim at a clear and coherent formalization of the process of conferring meaning to an image, which can be specialized to apply to concrete instantiations of computer vision algorithms.

[2] Quality spaces are related to the famous treatment of concepts in [3].

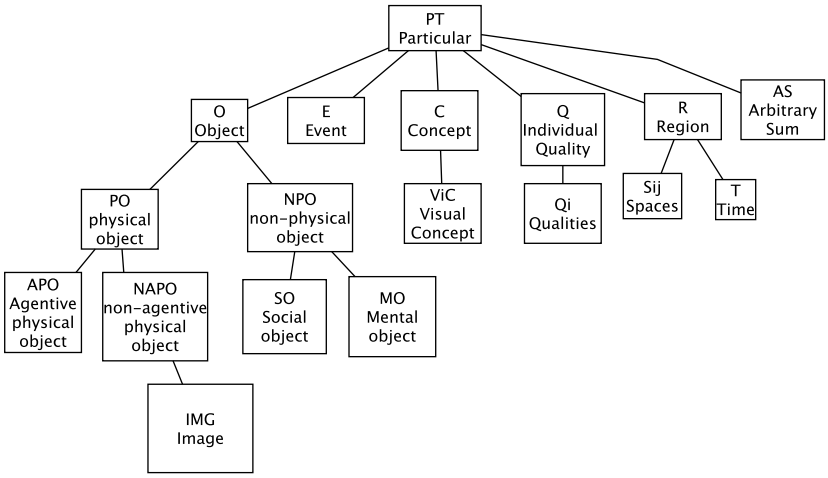

Fig. 1. Excerpt of DOLCE

We introduce the treatment in a discursive way, then at the end of this section, we will sum up the technicalities of our approach.

3.1 Visual concepts

We start by assuming a number of *visual concepts* $\text{VIC} = \{c_1, \ldots, c_n\}$, cf. Figure 2.1. They classify (parts of) images and express properties of objects that are visible in a broad sense. They may include qualities such as color, length, shape, but also concepts classifying objects, e.g. "a square", " a table", "a chair". As previously stated, we distinguish, among concepts, visual concepts as those concepts that classify representations of objects. Other kinds of concepts, instead, directly classify real objects as chairs. In other terms, we could say that the application of visual concepts to objects could be read as: "x looks like a chair" instead of "x is a chair". The point is to distinguish objects and visual representations of objects. The reason is that in developing an integrated approach to image understanding, we want to distinguish properties of an object that are transferrable to its representation and properties that are not. Moreover, there are qualities that we can ascribe by means of vision (e.g. color) and qualities that we can only ascribe through other types of knowledge (e.g. weight, or marital status).

a1 $\mathsf{IMG}(x) \rightarrow \mathsf{PO}(x)$
a2 $\mathsf{IMG}(x) \rightarrow \mathsf{APOS}(x) \vee \mathsf{POS}(x)$
d1 $\mathsf{hasContent}(x, y, t) \equiv_{def} \exists x' \mathsf{P}(x', x) \wedge CF(x', y, t)$

Axiom (a1) states that images are physical objects. Axiom (a2) states that images are to be split in *atomic positions* APOS and *general positions* POS:

atomic positions are the minimal parts of the image to which we can ascribe meaning, whereas POS contains the mereological sums of atomic positions plus the maximal part of the image, i.e. the full image itself. These constraints on the category of images can be made precise by means of a few axioms, and we omit the details for lack of space. The meaning of definition (d1) is that an image (i.e. a physical object) has content y if there is a part of the image that can be classified by the concept y at time t. The parts of an image are contained in the categories APOS and POS. For example, suppose that there are two parts x' and x'' of an image x such that x' gets classified as a cat, by means of the visual concept c, and x'' gets classified as a dog, by means of the visual concept d. We can conclude that image x has as a content both a cat and a dog. Definition (d1) uses the notion of *part* which in general is accounted for by the mereology of DOLCE-CORE [1]. For concrete applications, the notion of part has to be instantiated by means of a suitable segmentation of an image provided by computer vision techniques that single out the parts of the image (boxes, patches, etc.) that are relevant for a classification task. We shall discuss this point in more details in the next sections.

The crucial part in order to interface computer vision techniques and symbolic reasoning can be now expressed in the following terms: under which conditions can we assume that $\texttt{CF}(x, y, t)$, where x is (part of) an image and y is a visual concept, hold?

3.2 Basic and defined concepts

We approach this question by separating two types of visual concepts: *basic concepts* and *defined concepts*. The intuitive distinction between the two is the following: y is a *basic concept* iff $\texttt{CF}(x, y, t)$ is true because of a computer vision algorithm for classifying y-things that we run on x at time t; by contrast, y is a *defined concept* iff there is a definition (i.e. an if-and-only-if statement) of $\texttt{CF}(x, y, t)$ by means of other formulas in the visual theory.
The distinction between the two types of concepts is not absolute and it often depends on the choice of the language that we introduce in order to talk about images, on the classification tasks, on the available classification algorithms. For instance, "chair" is viewed as a basic property in case we associate it directly to a classifier of chairs. It can also be viewed as a defined concept, provided we define it, for instance, by writing a formula that says that something is classified as a chair iff it has four legs. In the latter case, strictly speaking, there is no classifier for chairs, just the one for classifying legs, and the classification of an image as a chair is obtained as a form of reasoning, i.e. it is *inferred*[3]. Therefore, we assume that the category of visual concept is partitioned into two sets: basic concepts $B = \{b_1, \ldots, b_m\}$ and defined concepts $D = \{d_1, \ldots, d_l\}$.

Moreover, we assume that basic concepts have to classify atomic positions:

[3] Given what just stated, the choice of which concepts should be considered basic may sound too arbitrary. Nonetheless, this choice is as arbitrary as any choice of the primitives of whatever ontology. In our case, we can at least appeal to a pragmatic justification.

a3 $\texttt{CF}(x, b, t) \rightarrow \textsc{Apos}(x)$

When introducing concepts such as d and c, we also intend to introduce the relevant constrains on the possible classifications. For instance, we want to force the fact that something that looks like a dog does not look like a cat. We label these constraints *incompatibility constraints*. As we have seen, an image may in principle contain the representation of a dog and of a cat in different areas. For this reason, the meaning of incompatibility constraints has to be expressed by stating that there is *at least* one part of the image that cannot be classified under two incompatible concepts, e.g. both as a cat and as a dog.

In general, we write incompatibility constraints on visual concepts as follows:

a4 $\exists z P(z, x)(\texttt{CF}(z, y, t) \rightarrow \neg\texttt{CF}(z, y', t))$

For practical purposes, one can select which parts of the image cannot be classified under incompatible concepts. For instance, in case one knows the possible dimensions of the image that are relevant for separating two visual concepts. Suppose that we label by means of a constant p the part of the image where we impose the constraint: $\texttt{CF}(p, d, t) \rightarrow \neg\texttt{CF}(p, c, t)$. The time parameter of the classification relations CF allows for possible reclassifications of images by different concepts, thus it may express the process of running different algorithms at different times. For instance, in case p is classified as a dog at time t $\texttt{CF}(p, d, t)$ and as a cat at time t' $\texttt{CF}(p, c, t')$, this may be caused for instance by two different algorithms that do not agree on the classification of p[4]. The incompatibility constraints exclude that *at the same time* a certain part of the image can be classified under incompatible concepts. In case we want to keep track of the information about which algorithm is responsible for which classification, we may add an explicit further parameter to the CF relation and assume a set of symbols that are labels for computer vision algorithms, e.g. $\texttt{CF}(x, y, t, a)$.

Moreover, we shall assume that ViC contains general n-ary concepts. The reason is that we want to interpret the classification of two parts of an image as related by means of an act of classification as well. For instance, in case we want to interpret the relation between two parts of an image, say x' and x'', in terms of the relation of *being above*, this is an act of classification that can be expressed by a formula $\texttt{CF}(x', x'', y, t)$ where the classification takes two arguments x' and x''. In general, we write $\texttt{CF}(\bar{x}, y, t)$ to state that the n-tuple of parts of image $\bar{x} = x_1, \ldots, x_n$ is classified by the n-ary concept y.

3.3 Visual theory

We present two definitions that formalize our approach. We introduce the following language based on first-order logic in order to talk about images. We label it *visual language*. The language includes the relevant predicates and the constants of DOLCE-CORE, plus the visual concepts. The category of visual concepts

[4] This point may also suggest a treatment of movement in time: in p there was a dog at time t and there is a cat at time t'. We leave this suggestion for future work, since we are focusing on images and not on videos.

shall be split into two classes, basic and defined concepts. We assume that VIC contains general n-ary concepts. Moreover, we assume two sets of individual constants $\text{APOS} = \{p_{a_1}, \ldots, p_{a_m}\}$ for atomic positions and $\text{POS} = \{p_1, \ldots, p_n, p_t\}$ for complex positions. Both sets are labels for parts of images so they are elements of IMG[5]. As we shall see, the constants for atomic positions should be enough to guarantee that we have the necessary number of constants to label the relevant positions. Moreover, POS contains the mereological sums of any atomic position, and we assume that p_t is the largest region (that is the full image).

Definition 1 (Visual language). *$\mathcal{VL}$ is a fragment of the language of first-order logic whose alphabet is the one of FOL plus the language of* DOLCE-CORE*, plus a given set of constants* VIC *for n-ary visual concepts and two sets of constants* $\text{APOS} = \{p_{a_1}, \ldots, p_{a_m}\}$ *and* $\text{POS} = \{p_1, \ldots, p_n, p_t\}$ *for positions in the image.*

The set VIC *is partitioned into two sets B and D:*

- *basic concepts* $B = \{b_1, \ldots, b_m\}$
- *defined concepts* $D = \{d_1, \ldots, d_l\}$

Once we have the visual language, the information concerning the possible meanings that we may associate to images are specified by defining a *visual theory*. The visual theory contains the axioms of DOLCE-CORE, a set $\mathcal{C}_\mathcal{T}$, that is a set of formulas that express general semantic constraints on visual concepts (e.g. dogs are animals), a set of incompatibility constraints $\mathcal{I}_\mathcal{T}$, and a set of definitions that relate basic concepts to defined concepts. The set of definitions, denoted by $\mathcal{D}_\mathcal{T}$, has to satisfy the following constraint. We want that every defined visual concept may be reducible to a (boolean) combination of basic concepts. A *definition* of a concept $y \in D$ is a statement of the form $\texttt{CF}(\bar{x}, y, t) \leftrightarrow \psi$, where ψ is a formula of $\mathcal{VL}$. We say that the concept c_1 *directly uses* the concept c_2 if c_2 appears on the right hand side of a definition of c_1. The relation *use* is the transitive closure of *directly use*.

Def For every $y \in D$, there exists a definition $\psi \in \mathcal{D}_\mathcal{T}$ such that every concept in ψ uses only basic concepts in B

Thus the visual theory is defined as follows:

Definition 2 (Visual theory). *$\mathcal{VT}$ is a set of first-order logic statements that includes the axioms of* DOLCE-CORE *and three sets of formulas: Semantic Constraints $\mathcal{C}_\mathcal{T}$, Definitions $\mathcal{D}_\mathcal{T}$ and Incompatibility Constraints $\mathcal{I}_\mathcal{T}$ such that:*

- *$\mathcal{D}_\mathcal{T}$ satisfies the constraint Def;*
- *a formula is in $\mathcal{I}_\mathcal{T}$ iff it is of the form $\exists z P(z, x)(\texttt{CF}(z, y, t) \rightarrow \neg\texttt{CF}(z, y', t))$ or $(\texttt{CF}(p, y, t) \rightarrow \neg\texttt{CF}(p, y', t))$, where $p \in \text{APOS} \cup \text{POS}$ is a constant of $\mathcal{VL}$.*

[5] We are identifying the positions in an image with parts of the image, so the parts of the image are also members of the category IMG.

The intended interpretations of $\mathcal{VT}$ are given by constraining the possible models. We assume that for each basic concept $b \in B$, there is a computer vision algorithm that classifies b-regions of the image: if z is a region of the image, $\theta_b(z) = 1$ if x is classified as a b, 0 otherwise. The domain of $\mathcal{VT}$ has to include individuals for all the relevant regions in the image. We have then to relate the regions of the image with the constants for positions of our visual language. The constants for atomic positions p_{a_i} in the visual language are then interpreted in regions of the image. The number of relevant regions in the image depends on the algorithm corresponding to the basic visual concepts, as we shall see in Section 4.1. Since in any case the set of regions extracted by means of computer vision is finite, we can ensure to associate to each region a constant in APos. Let $\{a_1, \ldots, a_n\}$ be the set of regions of an image, and $\mathcal{I}$ the interpretation of the constants of $\mathcal{VL}$, we force $\mathcal{I}(p_{a_i}) = a_i$ to be surjective, that is, every region is interpreted by a constant p_{a_i}. The question whether every other position in Pos should correspond to a region is more delicate. For instance, we have assumed that Pos is closed under mereological sum of positions. In general, we do not need to assume to be able to identify the region of image that corresponds to the mereological sum of positions. If we intend to do so, we can introduce the union of the regions. In what follows, the complex positions are inferred to exist from the basic ones, therefore they may be interpreted in abstract individuals of the domain instead of being associated to concrete regions of an image obtained by means of computer vision techniques.

We can force the following constraint on the models of $\mathcal{VT}$. Denote by p_x a variable that ranges over regions of images, we force that every atomic position is classified by a basic concept b iff the corresponding algorithm classifies the corresponding region accordingly.

C1 $\mathcal{M} \models \texttt{CF}(x, b, t)$ *iff* $\theta_b(p_x) = 1$

4 Application: a visual theory for geometric shapes

This example is intended to model a *folk* geometry of figures rather than the mathematical theory of polygons. We assume concepts such as being a quadrilateral, being an edge, being an angle. Moreover, we assume relational concepts such as `Touch` that is intended to express that two edges are touching in one of their extreme points. For a better readability, we write concepts in their predicative form: instead of writing $\texttt{CF}(x, \texttt{concept}, t)$, we write it by $\texttt{concept}(x, t)$.

The basic concepts are: $B = \{\texttt{Edge}(x, t), \texttt{Angle}(x, t), \texttt{Touch}(x, y, t)\}$. Since those are basic concepts, in order to check whether an image can be classified as an edge, we need to run a computer vision algorithm on the (part of) image x. By contrast, the other concepts are defined. For instance, polygons are here assumed to be just quadrilateral or trilateral. The set of semantic constraints $\mathcal{C}_\mathcal{T}$ is:

S1 $\texttt{EdgeOf}(x, y, t) \rightarrow \texttt{Edge}(x, t) \wedge \texttt{Polygon}(y, t)$
S2 $\texttt{AngleOf}(x, y) \rightarrow \texttt{Angle}(x, t) \wedge \texttt{Polygon}(y, t)$

S3 $\texttt{Touch(x,y,t)} \rightarrow \texttt{Edge}(x,t) \wedge \texttt{Edge}(y,t)$

Defined concepts and the set of definitions $\mathcal{D}_\mathcal{T}$ are the following. Recall that $\exists^n$ is the shortcut for "there exist exactly n". The set of definitions is then $\mathcal{D}_\mathcal{T}$:

D1 $\texttt{EdgeOf}(x,y,t) \leftrightarrow P(x,y) \wedge \texttt{Edge}(x,t)$
D2 $\texttt{AngleOf}(x,y,t) \leftrightarrow P(x,y) \wedge \texttt{Angle}(x,t)$
D3 $\texttt{PartOfFigure}(x,y,t) \leftrightarrow \texttt{EdgeOf(x,y,t)} \vee \texttt{AngleOf(x,y,t)}$
D4 $\texttt{Polygon}(x,t) \leftrightarrow \texttt{Quadrilateral}(x,t) \vee \texttt{Trilateral}(x,t)$
D4 $\texttt{Connected}(x,y,t) \leftrightarrow \exists z(\texttt{Edge}(z,t) \wedge \texttt{Touch}(x,z,t) \wedge \texttt{Touch}(z,y,t))$
D5 $\texttt{Trilateral}(x,t) \leftrightarrow \exists^3 y \texttt{EdgeOf}(y,x,t) \wedge \forall vw, \texttt{EdgeOf}(v,x,t) \wedge \texttt{EdgeOf}(w,x,t) \rightarrow \texttt{Connected}(v,w,t)$
D6 $\texttt{Quadrilateral}(x,t) \leftrightarrow \exists^4 y \texttt{EdgeOf}(y,x,t) \wedge \forall vw, \texttt{EdgeOf}(v,x,t) \wedge \texttt{EdgeOf}(w,x,t) \rightarrow \texttt{Connected}(v,w,t)$

Note that a number of incompatibility constraints can be inferred from the definitions in this case, e.g. $\exists x \texttt{Trilateral}(x,t) \rightarrow \neg \texttt{Quadrilateral}(x,t)$.

4.1 Verification of basic concepts by computer vision algorithms

The idea of the integrated system that we are developing mixes the computer vision layer and ontology-driven reasoning by using a two-fold approach. In the first step, diverse computer vision techniques serve to individuate and extract a set of interesting basic pattern regions in images that manifest patterns labelled as $\{a_1, \ldots, a_n\}$; in particular, we individuate *straight edges* and *angles* patterns, and we check whether these patterns share some geometrical relations, e.g. whether they are *touching* each other. We design then a set of elementary logic functions which serve to formally inject the patterns into the ontology reasoning. These functions correspond to basic concepts $\texttt{Edge}(x,t)$, $\texttt{Angle}(x,t)$, and $\texttt{Touch}(x,y,t)$. In the second step, the logic reasoning starts and individuates polygons in the image.

We briefly explain the techniques employed to individuate the *straight edges* and *angles* (thus creating the patterns $\{a_1, \ldots, a_n\}$), together with the functions corresponding to $\texttt{Edge}(x,t)$, $\texttt{Angle}(x,t)$ and $\texttt{Touch}(x,y,t)$. These are very standard techniques for the computer vision community and can be found in any image processing programming tool (in specific, we used MATLAB[6]).

Straight edges: The extraction of the edges (straight lines in the image) follows a two/step procedure: Sobel filtering followed by Hough transform. Sobel filtering [6] has been applied on the whole image; it basically consists in comparing adjacent pixels in a local neighborhood (a 3×3 patch) looking for substantial differences in the gray levels: in facts, an edge is assumed as a local and compact discontinuity which holds at least for three 8-connected pixels in the chromatic signals, and the Sobel filter enhances and highlights such discontinuities. In particular, the output of the filter is a binary mask, where the pixels labelled as 1 are edges, 0 otherwise. In addition, for the design of the filter, it is also possible

[6] See `http://goo.gl/MjA48F`.

to infer the orientation (in degrees) of the edge. The Hough transform [5] takes the binary mask produced by the Hough transform and looks for longer edges, whose minimum length can be given as input parameter. A detailed explanation of the algorithm is out of the scope for this work: in simple words, it is a voting approach where each edge pixel (and its orientation) votes for a straight line of a particular orientation and offset w.r.t the horizontal axis in the image space. The output of the algorithm is a set of coordinates indicating the x-y coordinates in the image space of the extrema of each edge, and each set for convenience is labelled as $\{a_1, \ldots, a_j\}$.

$\texttt{Edge}(x, t)$ corresponds then to a function $\theta_{\texttt{Edge}}(x)$ that takes a pattern of interest $a_i \in \{a_1, \ldots, a_n\}$ and gives 1 if the pattern is an edge (which is known by construction), 0 otherwise.

$\texttt{Touch}(x, y, t)$: Two edges are defined as touching each other if the closest distance between them occurs between two extrema of the two edges. In order to deal with the noise in the image and in the process of extracting the edges (that is, two edges which perceptually are touching in the image could be identified as separated by one or two pixels after the edge extraction) the extrema points are considered as touching even if they are close by few pixels, where this confidence can be quantized using a threshold. We can label the function that checks whether two edges are touching by $\theta_{\texttt{Touch}}$.

Angles: an angle is defined as the zone in which two edges are touching. For this reason, we decide to capture this visual information as a small squared patch, individuated by the set of coordinates of its corners in the image set, and each set is labelled for convenience as $\{a_j + 1, \ldots, a_n\}$.

$\texttt{Angle}(x, t)$ corresponds then to a function $\theta_{\texttt{angle}}$ that takes a pattern of interest $a_i \in \{a_1, \ldots, a_n\}$ and gives 1 if the pattern is an angle (which is known by construction), 0 otherwise.

The computer vision algorithms correspond to the verification of the basic concepts of $\mathcal{VT}$ via the constraints C1. For instance, if $\theta_{\texttt{angle}}(a_j) = 1$, then we force in our model $\mathcal{M}$, $\mathcal{M} \models \texttt{angle}(p_{a_j}, t)$, where p_{a_j} is an individual constant in $\mathcal{VL}$ that corresponds to the region a_j.

4.2 An example of classification by reasoning

We have seen that the classification of an angle is a matter of running a certain computer vision algorithm, that is, $\texttt{angle}(p_{a_j}, t)$ holds because of what we view as an act of perception. By contrast, in order to classify a quadrilateral, we need, in our example, to perform reasoning. `quadrilateral` is a defined concept, so in order to check whether a part of image y can be classified as a quadrilateral we use the definition of the concept, cf. D6. Thus, we need to check whether there are four parts of y that can be classified as edges of y (cf definition of `EdgeOf`, D1) that are moreover connected. Then, we need to use the definition of `connected`, cf D4. At this point, the definition of quadrilateral is reduced to a combination of basic concepts that can be checked by means of the corresponding computer vision algorithms. If the boolean combination of the outputs of the

computer vision algorithms – that is encoded by the definition of the concept `quadrilateral` – returns "true", then the part of image y is classified as a quadrilateral. Therefore, we can say that in this framework we can infer the presence of quadrilaterals instead of perceiving it.

5 Concluding remarks and open problems

We have provided a number of important elements for developing an integrated system for visual classification that uses both computer vision algorithms and the inferential capability of an ontological framework. We have placed our approach within the foundational ontology DOLCE in order to provide a clear explanation and a formalization of the process of conferring meaning to images, by means of the notion of classification under a visual concept. We have presented an apparently simple instantiation of our model to a simple ontology of polygons. The task of recognizing polygons seems straightforward for computer vision systems, but indeed it is not. Actually, most of the statistical pattern recognition classifiers' works are following the standard training/testing pipeline; in the training stage, a pool of labelled data is given to the classifier to individuate in the feature space a subspace which contains elements of a single class. In this scenario, the choice of the features is crucial: in practice, they should encode discriminant visual aspects of the objects to be recognized. This choice is very hard, and as a matter of fact, most of the standard computer vision approaches focus on distinguishing objects that are strongly different (car vs. motorbikes etc.) in which the visual cues are representative of visually dissimilar aspects of the objects (silhouette, color etc.). In our case, the task of individuating polygons with different numbers of vertices is not easy for computer vision, since usual cues neither perform any kind of counting, nor include any of the semantic relationships among vertices we were able to account for. A possible future step will be that of comparing our strategy with standard computer vision classification techniques, showing the importance of a mixed ontology/computer vision mechanism.

The robustness of our strategy strongly depends on the ability of the computer vision of recognizing elementary cues (such as the vertices in our example), since all the remaining is performed by a reasoning engine. As a matter of fact, the computer vision literature offers very robust strategies for extracting these kinds of features, while it is much weaker when it moves to higher level reasoning scenarios (in other words, it is more reliable in detecting that there are a set of pixels that move in the image, than in recognizing that those pixels individuate a car). For this reason we expect our idea to be of great impact for the computer vision community.

Another important direction for future work includes possible implementations of the present approach. It is easy to rephrase our treatment within a tractable fragment of first-order logic in order to ensure decidability of reasoning. For instance, it is possible to adapt our treatment in OWL, in order to achieve an implementation of a visual theory in Protégé. Unfortunately, this requires a number of restrictions on the formulas that we have used. Although this

direction is certainly of practical interest, we have preferred to present the treatment within a larger fragment of first order logic. The reason is that by focusing on a restricted fragment, we loose a significant part of the foundational ontology that is capable of providing a formalization of the mechanisms for approaching the semantic gap. We have instead chosen to use a powerful language to provide an expressive conceptualization of the interface between computer vision and symbolic reasoning, in order to present a clear formulation of the problem of the semantic gap.

Acknowledgments: This work is supported by the VisCoSo project grant, financed by the Autonomous Province of Trento through the "Team 2011" funding programme.

References

1. Stefano Borgo and Claudio Masolo. Foundational choices in dolce. In Steffen Staab and Ruder Studer, editors, *Handbook on Ontologies*. Springer, second edition, 2009.
2. Ivan Donadello and Luciano Serafini. Mixing low-level and semantic features for image interpretation - A framework and a simple case study. In *Computer Vision - ECCV 2014 Workshops - Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II*, pages 283–298, 2014.
3. Peter Gärdenfors. *Conceptual spaces - the geometry of thought*. MIT Press, 2000.
4. Céline Hudelot, Jamal Atif, and Isabelle Bloch. Fuzzy spatial relation ontology for image interpretation. *Fuzzy Sets and Systems*, 159(15):1929–1951, 2008.
5. John Illingworth and Josef Kittler. A survey of the hough transform. *Computer vision, graphics, and image processing*, 44(1):87–116, 1988.
6. Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *Solid-State Circuits, IEEE Journal of*, 23(2):358–367, 1988.
7. Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari. Wonderweb deliverable d18. Technical report, CNR, 2003.
8. Claudio Masolo, Laure Vieu, Emanuele Bottazzi, Carola Catenacci, Roberta Ferrario, Aldo Gangemi, and Nicola Guarino. Social roles and their descriptions. In *Proc. of the 6th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-2004)*, pages 267–277, 2004.
9. Daniele Porello, Francesco Setti, Roberta Ferrario, and Marco Cristani. Multiagent socio-technical systems: An ontological approach. In *Coordination, Organizations, Institutions, and Norms in Agent Systems IX - COIN 2013 International Workshops, COIN@AAMAS, St. Paul, MN, USA, May 6, 2013, COIN@PRIMA, Dunedin, New Zealand, December 3, 2013, Revised Selected Papers*, pages 42–62, 2013.
10. Wilfrid S. Sellars. Empiricism and the philosophy of mind. *Minnesota Studies in the Philosophy of Science*, 1:253–329, 1956.
11. Umberto Straccia and Giulio Visco. Dlmedia: an ontology mediated multimedia information retrieval system. In *Proceedings of the 2007 International Workshop on Description Logics (DL2007), Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June, 2007*, 2007.
12. Christopher Town. Ontological inference for image and video analysis. *Mach. Vis. Appl.*, 17(2):94–115, 2006.

13. Rong Zhao and William I Grosky. Negotiating the semantic gap: from feature maps to semantic landscapes. *Pattern Recognition*, 35(3):593–600, 2002.