Machine Learning Journal manuscript No.
(will be inserted by the editor)

Meta-Interpretive Learning from noisy images

Stephen Muggleton · Wang-Zhou Dai · Claude Sammut · Alireza Tamaddoni-Nezhad · Jing Wen · Zhi-Hua Zhou

Received: date / Accepted: date

Abstract Statistical machine learning is widely used in image classification. However, most techniques 1) require many images to achieve high accuracy and 2) do not provide support for reasoning below the level of classification, and so are unable to support secondary reasoning, such as the existence and position of light sources and other objects outside the image. This paper describes an Inductive Logic Programming approach called Logical Vision which overcomes some of these limitations. LV uses Meta-Interpretive Learning (MIL) combined with low-level extraction of high-contrast points sampled from the image to learn recursive logic programs describing the image. In published work LV was demonstrated capable of high-accuracy prediction of classes such as *regular polygon* from small numbers of images where Support Vector Machines and Convolutional Neural Networks gave near random predictions in some cases. LV has so far only been applied to noise-free, artificially generated images. This paper extends LV by a) addressing *classification noise* using a new noise-telerant version of the MIL system Metagol, b) addressing *atrribute*

S.H. Muggleton
Department of Computing, Imperial College London, London, UK
E-mail: s.muggleton@imperial.ac.uk

W-Z Dai and Z-H Zhou
LAMDA group, Nanjing University, Nanjing, China
E-mail: daiwz,zhouzh@nju.edu.cn

C. Sammut
School of Computer Science and Engineering, University of New South Wales, Sydney, Australia
E-mail: claude@cse.unsw.edu.au

A. Tamaddoni-Nezhad
Department of Computer Science, University of Surrey, UK
E-mail: a.tamaddoni-nezhad@surrey.ac.uk

J. Wen
School of Computer and Information Technology, Shanxi University,
E-mail: wjing@sxu.edu.cn

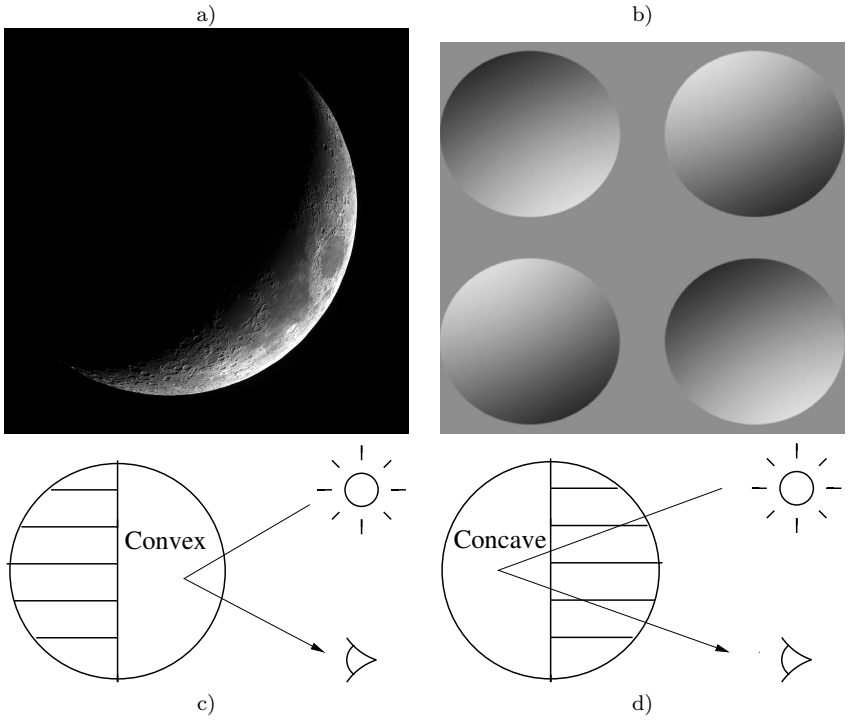
noise using primitive-level statistical estimators to identify sub-objects in real images, c) using a wider class of background models representing classical 2D shapes such as circles and ellipses, d) providing richer learnable background knowledge in the form of a simple but generic recursive theory of light reflection. In our experiments we consider noisy images in both natural science settings and in a RoboCup competition setting. The natural science settings involve identification of the position of the light source in telescopic and microscopic images, while the RoboCup setting involves identification of the position of the ball. Our results indicate that with real images the new noise-robust version of LV using a single example (i.e. one-shot LV) converges to an accuracy at least comparable to thirty-shot statistical machine learner on both prediction of hidden light sources in the scientific settings and in the RoboCup setting. Moreover, we demonstrate that a general background recursive theory of light can itself be invented using LV and used to identify ambiguities in the convexity/concavity of objects such as craters in the scientific setting and partial obscuration of the ball in the RoboCup setting.

1 Introduction

Galileo's *Siderius Nuncius* [15] describes the first ever telescopic observations of the moon. Using sketches of shadow patterns Galileo conjectured the existence of mountains containing hollow areas (i.e. craters) on a celestial body previously thought perfectly spherical. His reasoned description, derived from a handful of observations, relies on a knowledge of i) classical geometry, ii) straight line movement of light and iii) the Sun as an out-of-view light source. This paper investigates the use of Inductive Logic Programming (ILP) [33] to derive logical hypotheses, related to those of Galileo, from a small set of real-world images. Figure 1 illustrates part of the generic background knowledge used by ILP for interpreting object convexity in Experiment1 (Section 5.1).

Figure 1a shows an image of the crescent moon in the night sky, in which convexity of the overall surface implies the position of the Sun as a hidden light source beyond the lower right corner of the image. Figure 1b shows an illusion in which assuming a light source in the lower right leads to perception of *convex* circles on the leading diagonal. Conversely, a light source in the upper left implies their being *concave*. Figure 1c shows how interpretation of a *convex* feature, such as a mountain, comes from illumination of the *right* side of a convex object. Figure 1d shows that perception of a *concave* feature, such as a crater, comes from illumination of the *left* side. Figure 1e shows how Prolog background knowledge encodes a simple recursive definition of the reflected path of a photon.

This paper explores the phenomenon of knowledge-based perception using an extension of Logical Vision (LV) [10]. In the previous work LV was shown to accurately learn a variety of polygon classes from artificial images with low sample requirements compared to statistical learners. LV generates logical hypotheses concerning images using an ILP technique called Meta-Interpretive Learning (MIL) [32, 9].



e)
$light(X, X).$
$light(X, Y) :- reflect(X, Z),$
$light(Z, Y).$

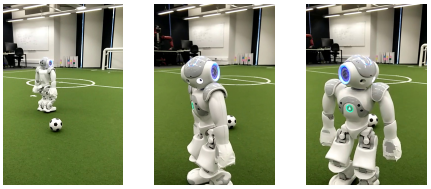
Fig. 1 Interpretation of light source direction: a) Waxing crescent moon (Credit: UC Berkeley), b) Concave/Convex illusion, c) Concave and d) Convex photon reflection models, e) Prolog recursive model of photon reflection

Contributions of this paper The main contributions of this paper are:

1. We describe a generalisation of LV [10], which is tolerant to both *classification noise* and *attribute noise*.
2. We show that even in the presence of noise in images (absent in artificial images in [10]) effective learning can be achieved from as few as one image.
3. We demonstrate that in all cases studied the combination of a logic-based learner with a statistical estimator requires far fewer images (sometimes one) to achieve accuracies requiring large numbers of images using statistical machine learning on its own.
4. We demonstrate that LV can use, as well as invent, generic background knowledge about reflection of photons in providing explanations of visual features.
5. We demonstrate that LV has potential in real application domains such as RoboCup.

RoboCup domain In Experiment 2 (Section 5.2) we investigate LV in the context of robotics. Figure 2 shows images from the RoboCup Soccer Standard Platform League [1]. This is a competition with five Aldebaran Nao robots on each team. They are placed on a 9m × 6m field, and operate autonomously to play soccer. The robots use cameras to detect the ball, field lines, goals and other robots. In Figure 2a, the ball can be seen distinctly, whereas Figures 2b and 2c the ball is partially occluded. The problem with recognising the ball is that it consists of several patches of black and white, but there are many other objects on the field that also contain white regions. However, background knowledge concerning the geometry of a sphere projected on a 2D plane guarantees a ball has a circular appearance. If three edge points can be found our approach can fit them to a circle and if that circle has the proportions of black and white pixels, the system concludes it is a ball.

The paper is organised as follows. Section 2 describes related work. The theoretical framework for LV is provided in Section 3. Section 4 describes the implementation of LV, including the recursive background knowledge for describing radiation and reflection of light. In Section 5 we describe experiments on 1) learning abstract definitions of polygons from artificial images, 2) predicting the light source direction and identification of ambigiuties in images of the moon and microscopic images of illuminated micro-organisms and 3) identifying the ball in the RoboCup domain. Finally, we conclude and discuss further work in Section 6.



(a) (b) (c)

Fig. 2 Robot's view of: a) another robot and ball clearly separated, b) the ball partially occluded by a robot, c) the ball within the bounds of a robot

2 Related work

Statistical machine learning based on low-level feature extraction has been increasingly successful in image classification [37]. However, high-level vision, involving interpretation of objects and their relations in the external world, is still relatively poorly understood [7]. Since the 1990s *perception-by-induction* [17] has been the dominant model within computer vision, where human perception is viewed as inductive inference of hypotheses from sensory data. The idea originated in the work of the 19th century physiologist Hermann von Helmholtz [44]. The approach described in this paper is in line with *perception-by-induction* in using ILP for generating high-level perceptual hypotheses by combining sensory data with a strong bias in the form of explicitly encoded

[1] http://www.tzi.de/spl/bin/view/Website/WebHome

background knowledge. Whilst Gregory [16] was one of the earliest to demonstrate the power of the Helmholtz's perception model for explaining human visual illusion, recent experiments [18] show Deep Neural Networks fail to reproduce human-like perception of illusion. This contrasts with results in Section 5.2, in which LV achieves analogous outcomes to human vision.

Early work in Computer Vision investigated the interaction between visual analysis, linguistic descriptions and geometric models [45,22]. In some such approaches visual illusions were identified by testing logical models of images for contradictions [2]. However, these techniques were based on preformulated models, and did not use machine learning augmented by background knowledge in the fashion described in this paper. Preformulated models are also used in more recent work to capture, for instance, the movement of a human being walking [19] or a hyperbolic curve involved in analysing images of penetrating radar [35]. However, these techniques lack the flexibility of our Logical Vision approach to combine a set of primitive models in a modular fashion to form a set of composite structured and re-useable models from an image.

Shape-from-shading [20,47] is a key computer vision technology for estimating low-level surface orientation in images. Unlike our approach for identifying concavities and convexities, shape-from-shading generally requires observation of the same object under multiple lighting conditions. By using background knowledge as a bias we reduce the number of images for accurate perception of high-level shape properties such as the identification of convex and concave image areas.

ILP has previously been used for learning concepts from images. For instance, in [6] object recognition is carried out using existing low-level computer vision approaches, with ILP being used for learning general relational concepts from this already symbolised starting point. Farid [13,14] adopted a similar approach, extracting planar surfaces from a 3D image of objects encountered by urban search and rescue robots and household objects, then using ILP to learn relational descriptions of those objects. By contrast, LV [10] uses ILP to provide a bridge from very low-level features, such as high contrast points, to high-level interpretation of objects. The present paper extends the earlier work on LV by implementing a noise-proofing technique, applicable to real images, and extending the use of generic background knowledge to allow the identification of objects, such as light sources, not directly identifiable within the image itself.

Various statistics-based techniques, making use of high-level vision, have been proposed for one- or even zero-shot learning [36,43]. They usually start from an existing model pre-trained on a large corpus of instances, and then adapt the model to data with unseen concepts. Approaches can be separated into two categories. The first exploits a mapping from images to a set of semantic attributes, then high-level models are learned based on these attributes [25, 29,36]. The second approach uses statistics-based methods, pre-trained on a large corpus, to find localized attributes belonging to objects but not the entire image, and then exploits the semantic or spatial relationships between the

Name	Metarule
PropObj1	$P(obj1) \leftarrow$
PropObj2	$P(obj2) \leftarrow$
PropLight	$P(light) \leftarrow$
Conjunct3	$P(x,y,z) \leftarrow Q(x,y,z), R(x,y,z)$
Chain3	$P(u,x,y) \leftarrow Q(u,x,z), R(u,z,y)$
Chain32	$P(u,x,y) \leftarrow Q(u,x,z), R(z,y)$
PrePost3	$P(x,y,z) \leftarrow Q(x,y), R(x), S(z)$
Pre2	$P(x) \leftarrow Q(x), R(x,y)$
Post2	$P(x,y) \leftarrow Q(x,y), R(y)$

Fig. 3 Metarules used in this paper. Uppercase letters P, Q, R, S denote existentially quantified variables. Lowercase letters u, x, y, and z are universally quantified.

attributes for scene understanding [21,26,12]. Unlike these approaches, we focus on one-shot from scratch, i.e. high-level vision based on just *very low-level features* such as high contrast points.

Machine learning is used extensively in robotics, mainly to learn perceptual and motor skills. Current approaches for learning perceptual tasks include Deep Learning and Convolutional Neural Networks [23,38]. The different approaches to vision in RoboCup can be seen in the SPQR team's use of convolutional neural networks [41] and the *ad hoc*, but effective method used by the 2016 SPL champions, B-Human [39]. This approach clearly depends on domain knowledge that has been acquired by the human designers. However, the approach described in this paper promises the possibility that similar knowledge could be acquired through machine learning.

3 Framework

The framework for LV is a special case of MIL.

Meta-Interpretive Learning Given background knowledge B and examples E the aim of a MIL system is to learn a hypothesis H such that $B, H \models E$, where $B = B_p \cup M$, B_p is a set of Prolog definitions and M is a set of *metarules* (see Figure 3). MIL [31,32,8,30,9] is a form of ILP based on an adapted Prolog meta-interpreter. A standard Prolog meta-interpreter proves goals by repeatedly fetching first-order clauses whose heads unify with the goals. By contrast, a MIL learner proves the set of all examples by fetching higher-order metarules (Figure 3) whose heads unify with the goals. The resulting meta-substitutions are saved, allowing them to be used to generate a hypothesised program which proves all the examples by substituting the meta-substitutions into corresponding metarules. Use of metarules and background knowledge helps minimise the number of clauses n of the minimal consistent hypothesis H and consequently the number of examples m required to achieve error below ϵ bound. [9] shows n dominates the upper bound for m[2].

[2] p predicates and M metarules $m \geq \frac{\mathrm{n}\ ln|M| + \mathrm{p}\ ln(3n) + ln\frac{1}{\delta}}{\epsilon}$

Logical Vision In LV [10], the background knowledge B, in addition to Prolog definitions, contains a set of one or more named images I. The examples describe properties associated with I.

4 Implementation

4.1 Noise tolerant Meta-Interpretive Learning

The MIL framework described in the previous section has been implemented in a system called $Metagol$ [31,32,8,30,9]. In this section we describe a noise tolerant version of $Metagol$ called $Metagol_{NT}$ [3]. The standard $Metagol$ implementation uses a modified Prolog meta-interpreter to backtrack through the space of hypotheses which prove all training examples. This strategy is consistent with an assumption of noise-free examples. Because of backtracking, standard methods for handling noise, such as accepting a user-defined maximum number of negative examples (used in the ILP systems Progol and Aleph), are inefficient for *Metagol* [4]. For this reason, a more efficient noise-handling method is required.

The noise tolerant version of $Metagol$ (i.e. $Metagol_{NT}$) used in this paper, finds hypotheses consistent with randomly selected subsets of the training examples and evaluates each resulting hypothesis on the remaining training set, returning the hypothesis with the highest score. The size of the training samples and the number of iterations (i.e. number of random samples) are user defined parameters. As shown in Algorithm 1, $Metagol_{NT}$ is implemented as a wrapper around $Metagol$ and returns the highest score hypothesis H_{max} learned from randomly sampled examples from E after n iterations. The sample size is controlled by $\nu = (k^+, k^-)$, where k^+ and k^- are the number of sampled positive and negative examples respectively, reflecting the noise level in the dataset.

4.2 Logical Vision

Our implementation of Logical Vision, called $LogVis$, is shown in Algorithm 2. The input consists of a set of images I, background knowledge B including both Prolog primitives B_p and metarules M, a set of training examples E of the target concept, $Metagol_{NT}$'s parameters ν and n.

The procedure of $LogVis$ is divided into two stages. The first stage is to extract symbolic background knowledge from images, which is done by the $visualAbduce$ function. By including abductive theories in $B_p \in B$, $visualAbduce$ can abduce points, lines, ellipses and even complex mid-level visual representations such as super-pixels (see Section 5.3). In our implementation, $visualAbduce$ can take logic rules, statistical models and functions from a computer vision toolbox as background knowledge, which provide visual primitives. This makes

[3] Available from https://github.com/metagol/Metagol_NT

[4] For example, a naive approach which ignores the label of the examples up to k times, has a time complexity of $O(\binom{m}{k})$, where m is the total number of examples.

Algorithm 1: $Metagol_{NT}(B, E, \nu, n)$

Input : Background knowledge B; Set of (noisy) examples E; Parameter about noise level ν and number of iterations n
Output: Hypothesis H_{max}

```
max_score = 0;
max = 1;
for each i ∈ [1, n] do
    /* Randomly select examples from E w.r.t. noise level ν          */
    Tr_i = randSample(E, ν);
    /* Leave the rest of examples for validation                     */
    Ts_i = E - Tr_i;
    /* Call Metagol and save the learned hypothesis in H_i           */
    H_i = learn(B, Tr_i);
    /* Evaluate the learned hypothesis H_i on validating set         */
    E_i = evaluate(B, H_i, Ts_i);
    if max_score < E_i then
        H_max = H_i;
        max_score = E_i;
    end
end
Return H_max;
```

Algorithm 2: $LogVis(I, B, E, \nu, n)$

Input : Training images I; Background knowledge B; Set of (noisy) examples E; Parameter about noise level ν and number of iterations n.
Output: Hypothesised logic program H.

```
/* Initialise the knowledge base of visual primitives               */
B_v = Φ;
for each image i ∈ I do
    /* Do visual abduction to get facts of visual primitives P      */
    P_i = visualAbduce(i, B);
    B_v = B_v ∪ P_i;
end
/* Call Metagol_NT to learn a model                                 */
Model = Metagol_NT(B ∪ B_v, E, ν, n);
Return Model;
```

LogVis flexible in learning many kinds of concepts. More details about visual abduction are introduced in Section 7.

The second stage of *LogVis* simply calls the noise-tolerant MIL system $Metagol_{NT}$ to induce a hypothesis for the target concept, as both abduced visual primitives B_v and training examples E from an image dataset can be noisy.

Visual abduction The target of visual abduction is to obtain symbolic interpretation of images for further learning. The abduced logical facts are groundings of primitives defined in the background knowledge B_p. For example, in order to learn the concept of a polygon one at least needs to extract points and edges from an image. When the data is noise-free, this can be done by sam-

pling high-contrast pixels from the image, such as the background knowledge about *edge_point* applied in [10].

However, for real images that contain a degree of noise, we can a include statistical model in *visualAbduce* and use it to implement a noise-robust version of *edge_point*. For example, in the *Protist* and *Moon* experiments of section 5, the *edge_point/1* calls a pre-trained statistical image background model which can categorise pixels into foreground or background points using Gaussian models or image segmentation.

Furthermore, we can use an abductive theory about shapes to abduce objects. For example, in real images many objects of interest are composed of curves and can be approximated by ellipses or circles. Therefore we can include background knowledge about them in *visualAbduce* to perform ellipse and circle abduction, as shown in Figure 4. The abduced objects will take the form *elps(Centre, Parameter)* or *circle(Centre, Radius)* where $Centre = [X, Y]$ is the shape's centre, $Parameter = [A, B, Tilt]$ are the axis lengths and tilting angle and *Radius* is the circle radius. The computational complexity of the abduction procedure is $O(rkn)$, where n is the number of `edge_points`, and k is the number of iteration of the ellipse fitting algorithm. r is the time required for resampling when the fitted object is not accurate enough, hence it is a constant that reflects the noise level of the input image.

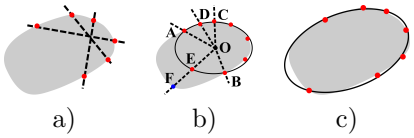


Fig. 4 Object detection: a) Sampled lines with edge points; b) Fitting of initial ellipse centred at O. Hypothesis tested using new edge points halfway between existing adjacent points. c) Revised hypothesis tested until hypothesis passes test.

In $LogVis$, background knowledge about visual primitives is implemented as logical predicates in a library, including basic geometrical concepts and extractors for low-level computer vision features such as the colour histogram and super-pixels. Users can implement their own background knowledge for visual abduction based on these primitives to address different kinds of problems flexibly.

5 Experiments

5.1 Experiment 1

In the first experiment (detailed report in [10]) we compared a noise-free variant of the $LogVis$ algorithm (refered to as LV_{Poly}) with statistics-based approaches on the task of learning simple geometrical concepts (see example images in Figure 5).

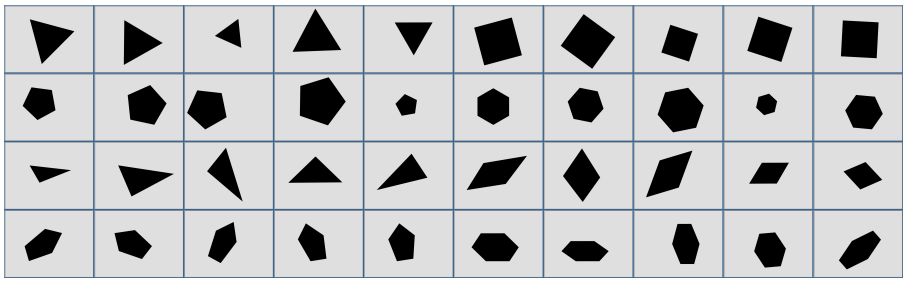


Fig. 5 Experiment 1 - examples of the concept *regular polygon* used in Experiment 1. The first two rows are positive examples and the second two rows negative.

Materials & methods We used Inkscape[5] to randomly generate 3 labelled image datasets for 3 polygon shape learning tasks respectively. Training sets contain 40 examples. For simplicity, the images are binary-coloured, each image contains one polygon. Target concepts are: 1) `triangle/1`, `quadrangle/1`, `pentagon/1` and `hexagon/1`; 2) `regular_poly/1` (regular polygon); 3) `right_tri/1` (right triangle). All the datasets were partitioned into 5-folds respectively, 4 of them were used for training and the remaining one is for testing, thus each experiment was conducted 5 times[6].

Results & discussion Table 1 compares the predictive accuracies of an implementation of LV_{Poly} versus several statistics-based computer vision algorithms. We used a popular statistics-based computer vision toolbox VLFeat [42] to implement the statistical learning algorithms. The experiments are carried with different kinds of features. Because the sizes of datasets are small, we used a support vector machine (libSVM [5]) as classifier. The parameters are selected by 5-fold cross-validation. The features we have used in the experiments are as follows: **HOG**, Histogram of Oriented Gradients [11], **Dense-SIFT**, Scale Invariant Feature Transform [28], **LBP**, Local Binary Pattern [34], **CNN**, Convolutional Neural Network (CNN) [40]. We also compare with a combinations of above feature sets (i.e. **C+d+L**). According to Table 1, given 40 training examples the prediction accuracies for LV_{Poly} are significantly better than other approaches.

5.2 Experiment 2

This subsection describes experiments comparing one-shot LV with multi-shot statistics-based learning[7]. In this experiments, we investigate the following null hypothesis:

Null hypothesis One-shot LV cannot learn models with accuracy comparable to thirty-shot statistics-based learning.

[5] http://inkscape.org
[6] Code at https://github.com/haldai/Logic-Vision
[7] Data and code at https://github.com/haldai/LogicalVision2

Table 1 Predictive accuracy of learning simple geometrical shapes from single object training sets of size 40.

ACC	tri	pen	hex	reg	r_tri
HOG	0.83 ± 0.04	0.73 ± 0.03	0.75 ± 0.07	0.63 ± 0.08	0.74 ± 0.04
DSIFT	0.82 ± 0.05	0.64 ± 0.04	0.71 ± 0.03	0.71 ± 0.05	0.77 ± 0.07
LBP	0.87 ± 0.05	0.67 ± 0.03	0.73 ± 0.03	0.65 ± 0.05	0.75 ± 0.05
CNN	0.91 ± 0.01	0.75 ± 0.00	0.84 ± 0.02	0.59 ± 0.06	0.85 ± 0.04
C+d+L	0.82 ± 0.01	0.76 ± 0.01	0.76 ± 0.01	0.64 ± 0.05	0.80 ± 0.04
LV_{Poly}	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.99 \pm 0.01}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$

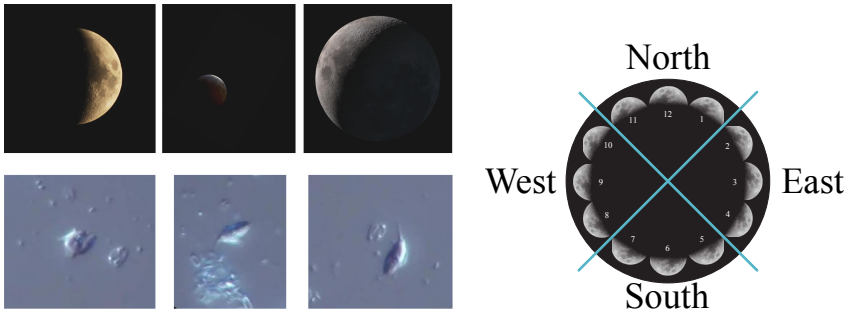


Fig. 6 Illustrations of *Moons* and *Protists* data: a) Examples of the datasets, b) Four classes for twelve light source positions

Materials We collected two real image datasets for the experiments: 1) **Protists** drawn from a microscope video of a *Protists* micro-organism, and 2) **Moons** a collection of images of the moon drawn from Google images. The instances in *Protists* are coloured images, while the images in *Moons* come from various sources and some of are grey-scale. For the purpose of classification, we generated the two datasets by rotating images through 12 clock angles [8]. Datasets consist of 30 images for each angle, providing a total of 360 images. Each image contains one of four labels as follows: $North = \{11, 12, 1\}$ clocks, $East = \{2, 3, 4\}$ clocks, $South = \{5, 6, 7\}$ clocks, and $West = \{8, 9, 10\}$ clocks. Examples of data and the labelling are shown in Fig 6. As we can see from the figure, there is high variance in the image sizes and colours.

Methods The aim is to learn a model to predict the correct category of light source angle from real images. For each dataset, we randomly divided the 360 images into training and test sets, with 128 and 232 examples respectively. To evaluate the performance, the models were trained by randomly sampling 1, 2, 4, 8, 16, 32, 64 and 128 images from the training set. The sequences of training and test instances are shared by all compared methods. The random partition of data and learning are repeated 5 times.

Logical Vision In the experiments, we used the grey intensity of both image datasets for LV. The hyper-parameter T in Algorithm 2 is set at 11 by val-

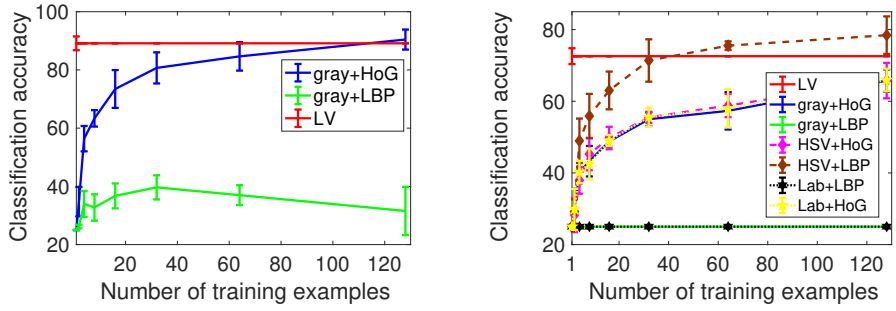
[8] Clock face angle between 12 and each hour position in {1..12}.

idating one-shot learned models on the rest of the training data. To handle image noise, we use a background model as the statistics-based estimator for predicate *edge_point/1*. When *edge_point([X, Y])* is called, a vector of colour distribution (which is represented by histogram of grey-scale value) of the 10×10 region centered at (X,Y) is calculated, then the background model is applied to determine whether this vector represents an edge point. The parameter of neighborhood region size 10 is chosen as a compromise between accuracy and efficiency after having tested it ranging from 5 to 20. The background model is trained from 5 randomly sampled images in the training set by providing the bounding box of the objects.

Statistics-based Classification The experiments with statistics-based classification were conducted in different colour spaces combined with various features. Firstly, we performed feature extraction to transform images into fixed length vectors. Next SVMs (libSVM [5]) with RBF kernel were applied to learn a multiclass-classifier model. Parameters of the SVM are chosen by cross validation on the training set. Like LV, we used grey intensity from both image datasets for the experiments. For the coloured *Protists* dataset, we transformed the images to **HSV** and **Lab** colour spaces to improve the performance. Since the image sizes in the dataset are irregular, during the object detection stage of LV, we used background models and computer graphics techniques (e.g. curve fitting) to extract the main objects and unified them into same sized patches for feature extraction. The sizes of object patches were 80×80 and 401×401 in *Protists* and *Moons* respectively. For the feature extraction process, we avoided descriptors which are insensitive to scale and rotation, instead we selected the luminance-sensitive features **HOG** and **LBP**. The Histogram of Oriented Gradient (HOG) [11] is known for its ability to describe the local gradient orientation in an image, and widely used in computer vision and image processing for the purpose of object detection. Local binary pattern (LBP) [34] is a powerful feature for texture classification by converting the local texture of an image into a binary number.

In the *Moons* task, LV and the compared statistics-based approach both used geometrical background knowledge for fitting circles (though in different forms) during object extraction. However, in the *Protists* task, the noise in images always caused poor performance in automatic object extraction for the statistics-based method. Therefore, we provided additional supervision to the statistics-based method consisting of bounding boxes labelling the position of the main objects in both training and test images during feature extraction. By comparison LV discovers the objects from raw images without any label information.

Results Figure 7a shows the results for *Moons*. Note that performance of the statistics-based approach only surpasses one-shot LV after 100 training examples. In this task, background knowledge involving circle fitting exploited by LV and statistics-based approaches are similar, though low-level features used by the statistics-based approach are first-order information (grey-scale gradients), which is stronger than the zeroth-order information (grey-scale value)



a) *Moons* b) *Protists*

Fig. 7 Classification accuracy on the two datasets.

```
clock_angle(A,B,C):-
    clock_angle1(A,B,D),
    light_source_angle(A,D,C).
clock_angle1(A,B,C):-
    highlight(A,B),
    clock_angle2(A),clock_angle3(C).
clock_angle2(obj1).
clock_angle3(light).
```

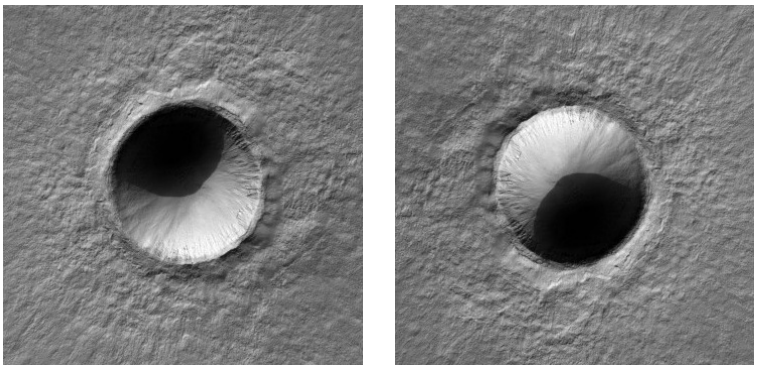
Fig. 8 Abductive program learned by LV: *clock_angle/3* denotes the clock angle from B (highlight) to A (object). *high_light/2* is a built-in predicate meaning B is the highlight part of A. *light_source_angle/3* is an abducible predicate and the learning target. With background knowledge about lighting and compare this program with Figure 9, we can interprete the invented predicate *clock_angle2* as *convex*, *clock_angle3* as *light_source_name*.

used by LV. Results on *Protists* are shown in Figure 7b. After 30+ training examples only one statistics-based approach outperforms one-shot LV. Since the statistics-based approaches have additional supervision (bounding box of main object) in the experiments, improved performance is unsurprising. The results of LV in Figures 7a and 7b are represented by horizontal lines. When the number of training examples exceeds one, LV performs multiple one-shot learning and selects the most frequent output (see Algorithm 2), which we found is always in the same equivalent class in LV's hypothesis space. This suggests LV learns the optimal model in its hypothesis space from a single example. The learned program is shown in Figure 8.

The results in Figure 7 demonstrate that *Logical Vision* can learn an accurate model using a single training example. By comparison, the statistics-based approaches require 40 or even 100 more training examples to reach similar accuracy, which refutes the null hypothesis. However, the performance of LV heavily relies on the accuracy of the statistical estimator of *edge_point/1*, because the mistakes of edge points detection will harm the shape fitting and consequently the accuracy of main object extraction. Unless we train a better *edge_point/1* classifier, the best performance of *LV* is limited as Figure 7 shows.

```
clock_angle(A,B,C):-
    clock_angle1(A,B,D),clock_angle4(A,D,C).
clock_angle1(A,B,C):-
    highlight(A,B),clock_angle2(A),clock_angle3(C).
clock_angle4(A,B,C):-
    light_source_angle(A,B,D),opposite_angle(D,C).
```

Fig. 9 Program learned by LV when concave objects are given as training examples.



a) Crater b) Flipped crater

Fig. 10 An image of a crater on Mars and the 180° rotated version. Credit: NASA/JPL/University of Arizona.

LV is implemented in SWI-Prolog [46] with multi-thread processing. Experiments were executed on a laptop with Intel i5-3210M CPU (2.50GHz), the time costs of object discovery are 9.5 seconds and 6.4 seconds per image on *Protists* and *Moons* dataset respectively; the average running time *Metagol* procedure is 0.001 second on both datasets.

Protists and *Moons* contain only convex objects. If instead we provide images with concave objects (such as Figure 10), LV learns a program such as Figure 9. Here the invented predicate *clock_angle2/1* can be interpreted as *concave* because its interpretation can be related to the appearance of *opposite_angle/2*.

Discussion: Learning ambiguity Figure 10 shows two images of a crater on Mars, where Figure 10b is a 180° rotated image of Figure 10a. Human perception often confuses the convexity of the crater in such images[9]. This phenomenon, called the *crater/mountain illusion*, occurs because human vision usually interprets pictures under the default assumption that the light is from the top of the image.

LV can use MIL to perform abductive learning. We show below that incorporation of generic recursive background knowledge concerning light enables LV to generate multiple mutually inconsistent perceptual hypotheses from real

[9] http://www.universetoday.com/118616/do-you-see-a-mountain-or-a-crater-in-this-picture/

```
clock_angle(O,H,A):-
    highlight(O,H),convex(O),light_source(L),
    light_source_angle(O,L,A).
clock_angle(O,H,A):-
    highlight(O,H),concave(O),light_source(L),
    light_source_angle(O,L,A1),opposite(A1,A).
```

Fig. 11 Interpreted BK learned by LV.

Abducibles
prim(convex/1). prim(concave/1). prim(light_source/1). prim(light_source_angle/3).
Compiled BK
% "obj1" is an object abduced from image, "obj2" is *% the brighter part of "obj1"; "observer" is the camera* contains(obj1,obj2). brighter(obj2,obj1). observer(observer). reflector(obj2). light_path(X,X). light_path(X,Y):-unobstructed(X,Z), light_path(Z,Y).
Interpreted BK
highlight(X,Y):- contains(X,Y),brighter(Y,X),light_source(L), light_path(L,R),reflector(R),light_path(R,O), observer(O).

Fig. 12 Background knowledge for learning ambiguity from images.

images. To the authors' knowledge, such ambiguous prediction has not been demonstrated previously with machine learning.

Recall the learned programs from Figure 8 and Figure 9 from the previous experiments. If we rename the invented predicates we get the general theory about lighting and convexity shown in Figure 11.

Now we can use the program as a part of interpreted background knowledge for LV to do abductive learning, where the abducible predicates and the rest of background knowledge are shown in Figure 12.

If we input Figure 10a to LV, it will output four different abductive hypotheses for the image, as shown in Figure 13[10]. From the first two results we see that, by considering different possibilities of light source direction, LV can predict that the main object (which is the crater) is either convex or concave, which shows the power of learning ambiguity. The last two results are even more interesting: they suggest that *obj2* (the highlighted part of the crater) might be the light source as well, which indeed is possible, though seems unlikely.[11]

[10] Code also at https://github.com/haldai/LogicalVision2

[11] The result can be reproduced and visualized by the example in Logical Vision 2 repository.

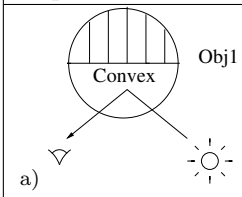
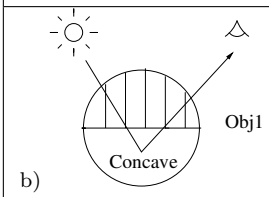
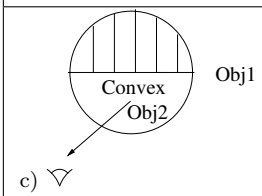
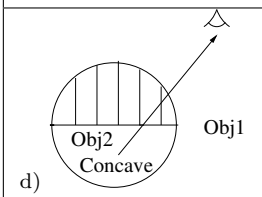
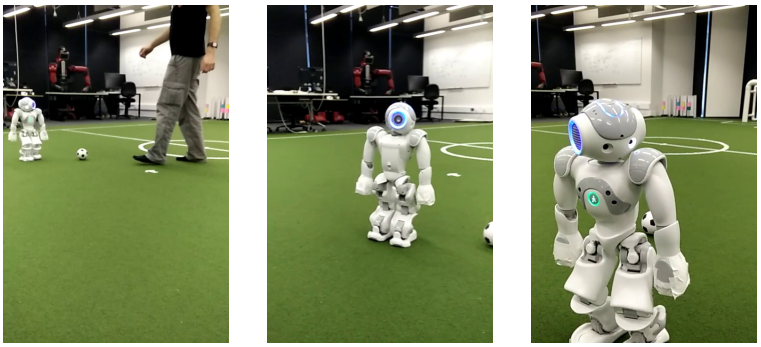
Depiction	Hypothesis
	light_source(light). light_source_angle(obj1,light,south). convex(obj1).
	light_source(light). light_source_angle(obj1,light,north). concave(obj1).
	light_source(obj2). light_source_angle(obj1,obj2,south). convex(obj1).
	light_source(obj2). light_source_angle(obj1,obj2,north). concave(obj1).

Fig. 13 Depiction of abduced hypotheses from Figure 10a.

5.3 Experiment 3

In this subsection we describe the experiments conducted on real images involving RoboCup [12] soccer where the task is to locate the football. We address this task in two stages: first we try to approximately locate the football in the image and then we use the model-driven technique of *Logical Vision* to abduce its location and shape. By doing this, one can estimate the size of the football, recognise occluded footballs and deduce depth information from the images.

[12] www.robocup.org



a b c

Fig. 14 Examples of football images: a) The football is clearly separated from other objects, b) part of the football is located outside of the image, c) the football is occluded by the robot.

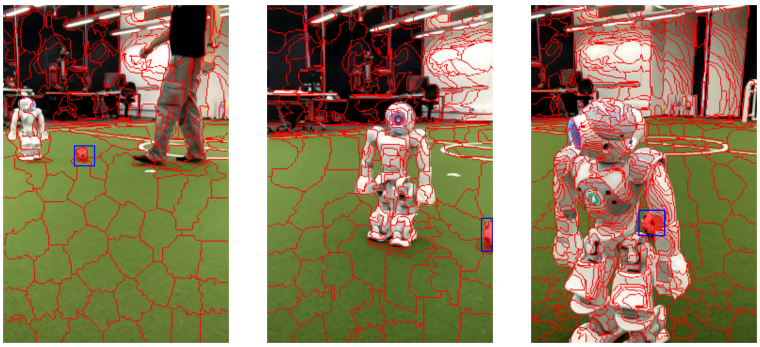
Dataset and task The dataset contains 377 colour images sampled from a video of the robot's camera view of the football field. As figure 14 shows, the scene of this dataset contains the green field, a robot, and a football. The original size of the images are 480×720. In this experiment they have been scaled into 240×360 to reduce the running time.

This task is more difficult than those in the previous experiments. The objects in the images are more complex and contain more noise. Therefore it is difficult to learn a hypothesis using simple primitives such as "edge_point". For example, the robot and football contain many edges so the original line sampling based abduction used by *Logical Vision* will become a large-scale combinatorial optimisation problem. Moreover, in 41 of the images the football is either occluded by or connected to other objects, and in 40 images there is no football at all.

To address the challenges, we consider a two-staged learning procedure. The first sub-task is to quickly find candidate locations of the footballs, which can reduce the search space of the fine grained football discovery. The second sub-task is to use *Logical Vision* to abduce the location and shape of the football from the candidate positions.

For the first sub-task, we use a super-pixel algorithm [1] to segment the images into small regions, which can serve as primitives for estimating the location of football. Super-pixel algorithms are able to group pixels into atomic regions that capture image redundancy, greatly reducing the complexity of subsequent image processing tasks. The super-pixel algorithm implementations we used are OpenCV_contrib[13] [3]. The tuned parameter is the size of each super-pixel, which ranges from 10 to 30 with step size 5. During data transformation, we use the football bounding boxes shipped with original images to label the super-pixels: those which have 95% area inside of bounding box of footballs (which is the label information in original data) are labelled as positive ex-

[13] https://github.com/opencv/opencv_contrib



a b c

Fig. 15 Super-pixel segmented data of the images in Figure 14, where the blue boxes are the original bounding boxes of the images, the super-pixels filled with red colour are the positive super-pixels according to the bounding boxes. Note that in b) and c), although the footballs have been split into multiple super-pixels, they are all labelled as positive examples.

amples with predicate "ball_sp". The rest are labelled as negatives. Examples from the dataset are shown in Figure 15. The second sub-task, model-driven football abduction, directly takes "ball_sp" and an abductive theory as input and outputs the circle parameters (centre and radius), where "ball_sp" should be the result produced by the classification model learned in the first stage.

Experiment: Football super-pixel classification This experiment is related to the first sub-task described above, i.e. locating the football from super-pixel segmented images. In this experiment we compare the performance of $Metagol_{NT}$ versus a statistical learner (we choose the CART algorithm [4][14]) and investigate the same null hypothesis used in Section 5.2.

Materials and methods In this experiment we use the super-pixel dataset as described above. Each super-pixel is regarded as a symbolic object in the background knowledge. We extract some basic properties, such as size, location and colour distribution as features. The colour distribution is represented by the proportion of white, grey, black and green pixels inside a super-pixel, which is identified by **Lab** values of the pixels. Moreover, we exploit the neighbourhood relationship between super-pixels, which is represented by the "next_to/2" predicate[15].

In this experiment we randomly sample 128 images for the training and the remaining 249 images for testing. Similar to the *Protists* and *Moons* experiments in Section 5.2, we randomly sample 1, 2, 4, 8, 16, 32, 64, 128 images from the training set for learning the classification model. Random data partitioning is performed 5 times. The positive training examples (both for the statistical

[14] CART was chosen since it is efficient and provides human-comprehensible output comparable to logic programs and execution of decision trees within the Robocup environment is sufficiently efficient (under 1/30th of a second) for localisation and decision making

[15] Dataset located at https://github.com/haldai/LogicalVision2/tree/master-2.1/data

learner and the relational learner) are football super-pixels from each of 1, 2, 4, 8, 16, 32, 64, 128 images and the same number of negative examples (i.e. non-football super-pixels) are randomly sampled from the same set of training images. Similarly, for the test data the negative examples are randomly sampled from non-football super-pixels in the test images. For relational learning (i.e. $Metagol_{NT}$), background predicates *mostly_white*/1, *partly_white*/1, *mostly_black*/1, *partly_black*/1, etc were defined based on the colour distribution of super-pixels. For example the following background definitions describe a super-pixel which is mostly white or partly white:

```
mostly_white(S):- white(S, P), P > 0.6.
partly_white(S):- white(S, P), P > 0.4, P =< 0.6.
```

The background knowledge for the relational learner also includes the neighbourhood relationship between super-pixels, i.e. "next_to/2" predicates.

In this experiment the following parameters were used for the relational learner, i.e. $Metagol_{NT}(B, E, \nu, n)$ in Algorithm 1. In addition to the above mentioned background knowledge, B includes the $Pre2$ and $Post2$ Meta-rules from Fig 3.

E is the set of positive and negative training examples as described above. The size of randomly selected training examples $Tr_i \subset E$ in each iteration i of Algorithm 1 and the number of iterations n can be set according to the expected degree of noise. Given that the expected error rate in the training data is not known in this problem, we choose an extreme case where Tr_i contains one randomly selected positive example (and one or two randomly selected negative examples) in different experiments. The number of iterations n was set to the number of positive examples in E.

For the statistics-based learner we use the CART decision tree algorithm [4]. The goal is to create a model that predicts the value of a target variable based on splitting the feature space. We choose CART as the compared method because we want to ensure the statistical model uses the same features as the relational model. Since the number of features, i.e. the green/white/grey/black pixel proportions, is relatively small, it is natural to choose a decision tree as the statistical learner. The maximum number of splits, is automatically selected by 5-fold cross validation on the training data.

A second reason for the choice of decision trees is efficiency of execution. The robots in RoboCup soccer must operate in real-time, which means that all vision, localisation, decision making, localisation and locomotion tasks must be completed in the time it takes to capture the next camera frame, typically 1/30th of a second. Thus, the classifier in the vision system must be extremely efficient to execute. A decision tree, with only a few comparisons leading to a decision in the leaf node, satisfies these stringent timing requirements.

Results Figure 16 compares the predictive accuracy of the relational learner ($Metagol_{NT}$) vs the statistics-base learner (CART). As shown in the figure,

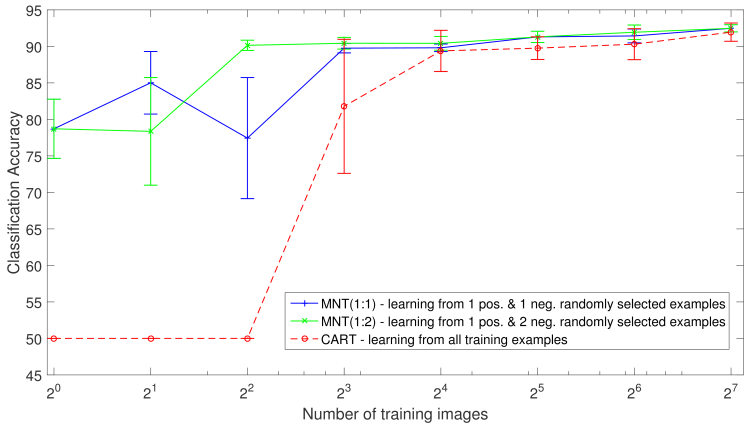


Fig. 16 Accuracy of $Metagol_{NT}$ vs CART in the task of football super-pixel classification.

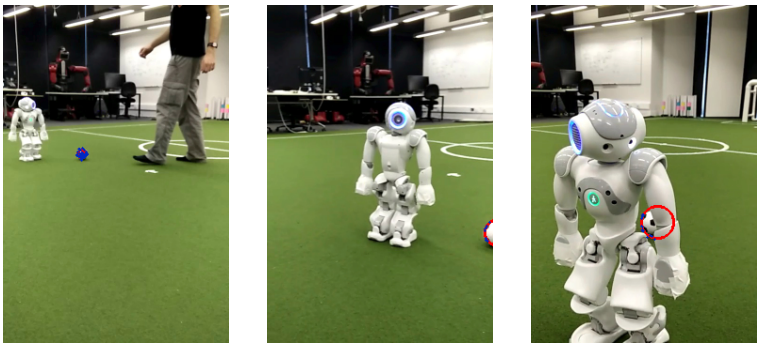
$Metagol_{NT}$ achieves consistently higher accuracy than CART with the accuracy difference particularly high for small numbers of training examples. An example of the hypotheses found by the relational learner is as follows:

```
ball_sp(A):- partly_white(A), ball_sp_1(A, B).
ball_sp_1(A,B):- next_to(A, B), mostly_green(B).
```

Model-driven football abduction After narrowing down the candidate location of the football, *Logical Vision* is able to exploit geometrical background knowledge to perform model-driven abduction of the football's exact shape and position (i.e. its centre and radius as a circle). This is important in robotic football games since the robot can use this information to infer the distance between itself and the football. More importantly, by modelling the football with a circle, the robot can figure out the occlusion of the football by other robots and choose approriate actions accordingly. We apply *Logical Vision* with an abductive theory for this task, whose abducible is "football/3". To sample edge points, *Logical Vision* draws random straight lines inside a super-pixel and its neighbourhood to return the points associated with a colour transition. Examples of football abduction are shown in Figure 17.

6 Conclusions and further work

Human beings often learn visual concepts from single image presentations (so-called one-shot-learning) [24]. This phenomenon is hard to explain from a standard Machine Learning perspective, given that it is unclear how to estimate any statistical parameter from a single randomly selected instance drawn from an unknown distribution. In this paper we show that learnable generic logical background knowledge can be used to generate high-accuracy logical hypotheses from single examples. This compares with similar demonstrations



a b c

Fig. 17 Ball abduction results of the images in Figure 14. The blue points are the "edge_points" sampled by *Logical Vision*, the red curves are the abduced circles.

concerning one-shot MIL on string transformations [27] as well as previous concept learning in artificial images [10]. The experiments in Section 5 show that the LV system can accurately identify the position of a light source from a single real image, in a way analogous to scientists such as Galileo, observing the moon for the first time through a telescope or Hook observing micro-organisms for the first time through a microscope. In Section 5.2 we show that logical theories learned by LV from labelled images can also be used to predict concavity and convexity predicated on the assumed position of a light source. Section 5.3 shows how LV can be used effectively in real-time robot vision. Ball recognition in robot soccer is challenging because the ball is frequently occluded by other robots and the similarity in colours of the ball, robots and field lines makes the ball difficult to distinguish.

We have studied LV's failure cases carefully. The main reason causing misclassification is the noise in images. The noise can cause misclassifications of edge_point/1 since it is implemented with statistical models. The mistakes of edge_point detection will further affect the edge detection and shape fitting. As a result, the accuracy of the main object extraction is limited by both the noise level in input images and the power of statistical model of edge_point/1. Therefore, LV will fail too since the wrongly extracted objects are its inputs. However, if we train stronger models for detecting edge_points, the accuracy of LV will not increase either.

In further work we aim to investigate broader sets of visual phenomena which can naturally be treated using background knowledge. For instance, the effects of object obscuration; the interpretation of shadows in an image to infer the existence of out-of-frame objects; the existence of unseen objects reflected in a mirror found within the image. All these phenomena could possibly be considered in a general way from the point of view of a logical theory describing reflection and absorption of light, where each image pixel is used as evidence of photons arriving at the image plane. In this further work we aim to compare our approach once more against a wider variety of competing methods.

The authors believe that LV has long-term potential as an AI technology with the potential for unifying the disparate areas of logical based learning with visual perception.

References

1. R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.
2. H.G. Barrow and J.M. Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17:75–116, 1981.
3. G. Bradski. Opencv library. http://opencv.org/, 2000.
4. L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.
5. C-C. Chang and C-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.
6. A.G. Cohn, D.C. Hogg, B. Bennett, A. Galata, D.R. Magee, and P. Santos. In *Cognitive vision: Integrating symbolic qualitative representations with computer vision*, pages 221–246. Springer, Berlin, 2006.
7. D. Cox. Do we understand high-level vision? *Current opinion in neurobiology*, 25:187–193, 2014.
8. A. Cropper and S.H. Muggleton. Logical minimisation of meta-rules within meta-interpretive learning. In *Proceedings of the 24th International Conference on Inductive Logic Programming*, pages 65–78. Springer-Verlag, 2015. LNAI 9046.
9. A. Cropper and S.H. Muggleton. Learning higher-order logic programs through abstraction and invention. In *Proceedings of the 25th International Joint Conference Artificial Intelligence (IJCAI 2016)*, pages 1418–1424. IJCAI, 2016.
10. W-Z Dai, S.H. Muggleton, and Z-H Zhou. Logical Vision: Meta-interpretive learning for simple geometrical concepts. In *Late Breaking Paper Proceedings of the 25th International Conference on Inductive Logic Programming*, pages 1–16. CEUR, 2015.
11. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 13rd IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 886–893, San Diego, CA, 2005. IEEE Computer Society.
12. K. Duan, D. Parikh, D.J. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition*, pages 3474–3481, Providence, RI, 2012. IEEE Computer Society.
13. R. Farid and C. Sammut. Plane-based object categorisation using relational learning. *Machine Learning*, 94:3–23, 2014.
14. R. Farid and C. Sammut. Region-based object categorisation using relational learning. In *PRICAI 2014: Trends in Artificial Intelligence*, volume 8862 of *LNAI*, pages 1106—1114. Springer Verlag, 2014.
15. Galileo Galilei. *The Herald of the Stars*. 1610. English translation by Edward Stafford Carlos, Rivingtons, London, 1880; edited by Peter Barker, Byzantium Press, 2004.
16. R.L. Gregory. *Concepts and Mechanics of Perception*. Duckworth, London, 1974.
17. R.L. Gregory. *Eye and Brain: The Psychology of Seeing*. Oxford University Press, Oxford, 1998.
18. D. Heath and D. Ventura. Before a computer can draw, it must first learn to see. In *Proceedings of the 7th International Conference on Computational Creativity*, pages 172–179, 2016.
19. D Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1:5–20, 1983.
20. B.K.P. Horn. *Obtaining shape from shading information*. MIT Press, 1989.
21. R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *Proceedins of the 29th IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564, Las Vegas, NV, 2016. IEEE Computer Society.

22. D.A. Huffman. Impossible objects as nonsense sentences. In B. Meltzer and D. Michie, editors, *Machine Intelligence 6*, pages 295–323. Edinburgh University Press, Edinburgh, 1971.
23. A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106—1114, 2012.
24. B.M Lake, R. Salakhutdinov, J. Gross, and J.B. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, pages 2568–2573, 2011.
25. C.H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014.
26. Z. Li, E. Gavves, T. Mensink, and C.G.M. Snoek. Attributes make sense on segmented objects. In *Proceedings of 13th European Conference on Computer Vision Part IV*, pages 350–365, Zurich, Switzerland, 2014. Springer.
27. D. Lin, E. Dechter, K. Ellis, J.B. Tenenbaum, and S.H. Muggleton. Bias reformulation for one-shot function induction. In *Proceedings of the 23rd European Conference on Artificial Intelligence (ECAI 2014)*, pages 525–530, Amsterdam, 2014. IOS Press.
28. D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
29. T. Mensink, J.J. Verbeek, and G. Csurka. Learning structured prediction models for interactive image labeling. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition*, pages 833–840, Colorado Springs, CO, 2011. IEEE Computer Society.
30. S.H. Muggleton, D. Lin, J. Chen, and A. Tamaddoni-Nezhad. Metabayes: Bayesian meta-interpretative learning using higher-order stochastic refinement. In Gerson Zaverucha, Vitor Santos Costa, and Aline Marins Paes, editors, *Proceedings of the 23rd International Conference on Inductive Logic Programming (ILP 2013)*, pages 1–17, Berlin, 2014. Springer-Verlag. LNAI 8812.
31. S.H. Muggleton, D. Lin, N. Pahlavi, and A. Tamaddoni-Nezhad. Meta-interpretive learning: application to grammatical inference. *Machine Learning*, 94:25–49, 2014.
32. S.H. Muggleton, D. Lin, and A. Tamaddoni-Nezhad. Meta-interpretive learning of higher-order dyadic datalog: Predicate invention revisited. *Machine Learning*, 100(1):49–73, 2015.
33. S.H. Muggleton, L. De Raedt, D. Poole, I. Bratko, P. Flach, and K. Inoue. ILP turns 20: biography and future challenges. *Machine Learning*, 86(1):3–23, 2011.
34. T. Ojala, M. Pietikainen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.
35. G.R. Olhoeft. Maximizing the information return from ground penetrating radar. *Journal of Applied Geophysics*, 43:175–187, 2000.
36. M. Palatucci, D. Pomerleau, G. Hinton, and T.M. Mitchell. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems 22*, pages 1410–1418. Curran Associates Inc., 2009.
37. S.S. Rautaray and A. Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43:1–54, 2015.
38. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.
39. T. Rofer, T. Laue, J. Richter-Klug, and F. Thielke. B-Human Team Description for RoboCup 2016, 2016. http://www.robocup2016.org/media/symposium/Team-Description-Papers/StandardPlatform/RoboCup_2016_SPL_TDP_B-Human.pdf.
40. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, 2015.
41. V. Suriani, D. Albani, A. Youssef, F. Del Duchetto, A. Nania, D.D. Bloisi, L. Iocchi, and D. Nardi. SPQR RoboCup 2016 Standard Platform League Team Description Paper, 2016. http://www.robocup2016.org/media/symposium/Team-Description-Papers/StandardPlatform/RoboCup_2016_SPL_TDP_SPQR.pdf.

42. A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.
43. O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29*, pages 3630–3638. MIT Press, 2016.
44. H. von Helmholtz. *Treatise on Physiological Optics Volume 3*. Dover Publications, New York, 1962. Originally published in German in 1825.
45. D.L. Waltz. Understanding scene descriptions as event simulations. In *Proceedings of the 18th annual meeting on Association for Computational Linguistics*, pages 7–11. Association for Computational Linguistics, 1980.
46. J. Wielemaker, T. Schrijvers, M. Triska, and T. Lager. SWI-Prolog. *Theory and Practice of Logic Programming*, 12(1-2):67–96, 2012.
47. R. Zhang, P.S. Tai, J.E. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8):670–706, 1999.