Snakes in Trees: An Explainable Artificial Intelligence Approach for Automatic Object Detection and Recognition

Joanna Isabelle Olszewska
School of Computing and Engineering, University of the West of Scotland, U.K.

Keywords: Explainable Artificial Intelligence, Explainable by Design, Computer Vision, Machine Vision, Smart Cities, Industry 4.0, Intelligent Systems, Decision Tree, Snake, Active Contours, Recursive Algorithm, Unsupervised Labeling, Semantic Tag, Automatic Image Annotation.

Abstract: Nowadays, the development of smart cities boosts the development of innovative IT technologies based on Artificial Intelligence (AI), such as intelligent agents (IA), which themselves use new algorithms, complex software, and advanced systems. However, due to their expanding number and range of applications as well as their growing autonomy, there is an increased expectation for these intelligent technologies to involve explainable algorithms, dependable software, trustworthy systems, transparent agents, etc. Hence, in this paper, we present a new explainable algorithm which uses snakes within trees to automatically detect and recognize objects. The proposed method involves the recursive computation of snakes (aka parametric active contours), leading to multi-layered snakes where the first layer corresponds to the main object of interest, while the next-layer snakes delineate the different sub-parts of this foreground. Visual features are extracted from the regions segmented by these snakes and are mapped into semantic concepts. Based on these attributes, decision trees are induced, resulting in effective semantic labeling of the objects and the automatic annotation of the scene. Our computer-vision approach shows excellent computational performance on real-world standard database, in context of smart cities.

1 INTRODUCTION

According to the United Nations (UN), more than half of the World population currently lives in urban areas, and this trend is only rising (Bhattacharya et al., 2020). Therefore, there is a need to rethink cities in efficient and modern ways, leading to smart cities, which are defined as urban areas that create sustainable economic development and high quality of life by excelling in six key areas, namely, economy, mobility, environment, people, living, and government (Montemayor et al., 2015).

In smart cities, traditional infrastructures as well as new services are merged, coordinated, and integrated using innovative digital technologies (Batty et al., 2012). Hence, IT technologies such Artificial Intelligence (AI)-based systems are a cornerstone aspect of smart cities, with AI acting as a catalyst for areas such as smart urban modeling, intelligent infrastructures, smart transportation, smart governance, sustainability, smart education, and smart health solutions, to name a few (Bhattacharya et al., 2020).

In particular, computer-vision-based intelligent systems contribute to smart city’s applications such as smart surveillance (Ryabchikov et al., 2020), smart security (Khan et al., 2019), smart traffic management (Gupta and Sundar, 2020), and smart mobility (Fortes et al., 2021). Intelligent vision systems can thus improve people’s quality of life by helping drivers with finding vacant parking space (Bravo et al., 2013) or aiding visually impaired to safely travel around the smart city (Nasralla et al., 2019).

Intelligent vision systems are also embedded in a variety of intelligent agents (Olszewska, 2020) such as inspection robots (Sui, 2021), cleaning robots (Narang et al., 2021), etc. They can also be integrated into mobile ground robots for assistive operations in public spaces (Grzeskowiak et al., 2021) or into unmanned aerial vehicles (UAV) for data collection through the smart city (Shirazi et al., 2020).

All these computer-vision-based applications involve automatic object detection (Chen et al., 2021). Indeed, automatic object detection is used for object recognition (Wang et al., 2021), scene recognition (Zeng et al., 2020), activity recognition (Mliki et al., 2021).
2020), object tracking (Li et al., 2021), or automated image annotation (Zhang et al., 2012), which in turn can be applied to pedestrian detection and surveillance (An et al., 2021), car detection and annotation (Li et al., 2020), license plate recognition (Huang et al., 2021), robot navigation (Lin et al., 2021), and many more applications (Calzado et al., 2018), (Olszewska, 2018).

Therefore, the development of transparent and explainable artificial intelligence algorithms for intelligent vision systems (Olszewska, 2021) deployed in smart cities is of prime importance, due to the growing concern of citizens about, on one hand, their data privacy and security (Chourabi et al., 2012), and, on the other hand, the behaviour of the surrounding intelligent agents (Zhang et al., 2017).

Over the last decade, computer vision systems have been more and more relying on machine learning, and especially on deep learning (Camacho et al., 2021), which is a very popular and efficient approach. However, the use of deep learning involves most of the time very large and unbiased training datasets (Kishida et al., 2021) with high-resolution images (Wang et al., 2021), time and energy-consuming processes (Liu et al., 2021), resource-intensive computational power (Prakash et al., 2020), with associated, extensive costs (Chourabi et al., 2012) as well as sophisticated equipment (Namiki et al., 2021); all these demands being not always available in real-world conditions. Besides, deep learning is not currently considered as an explainable machine-learning approach (Gunning et al., 2019).

Since explainable artificial intelligence (XAI) is very important for users (Ferreira, J. J. and Monteiro, M. S., 2020), (Wilding et al., 2020) and regulators (Winfield et al., 2021), (Prestes et al., 2021), explainability becomes a ‘sought-after’ non-functional requirement (Kohli et al., 2019) of AI-based systems such as intelligent vision systems (Olszewska, 2019b).

Explainability can thus be assessed by internal algorithmic auditing (Raji et al., 2020), software testing (Black et al., 2022), and/or by verification and validation of the intelligent system (Corso et al., 2021).

Explainability can be also addressed at an earlier stage of the intelligent-vision system development (Olszewska, 2019a), i.e. during the system design (Bryson and Winfield, 2017) and its algorithm design (Mendling et al., 2021), leading to XAI by Design (Kearns and Roth, 2020).

Thus, in this work, we propose a novel, explainable-by-design AI-based algorithm for intelligent vision systems. Our algorithm consists mainly on the recursive computation of both the computer-vision method called snakes and the machine-learning-based decision trees.

Indeed, on one hand, object-of-interest’s shape (Samani et al., 2021) and closed contours (Funke et al., 2021) are very important visual feature for object detection and recognition (Lv et al., 2021). Therefore, computer-vision techniques such as active contours (Yezzi et al., 2019) are an efficient and explainable method to locate, segment, and track an object. In particular, we have selected the active-contour method known as ‘snake’ (Muralidhar et al., 2010), which automatically computes parametric active contours to delineate the visual-object shape, since snakes ally explainability with excellent detection performance (Olszewska, 2017).

On the other hand, we have adopted decision trees (DT), which are considered as the most explainable approach to machine learning (Gunning et al., 2019). Furthermore, decision trees are efficient for visual object detection and recognition applications (Nowozin et al., 2011), (Olszewska, 2015b).

Thence, our explainable-by-design algorithm can perform object detection and recognition in both static or dynamic scenes, depending of the type of input data - still image (Li et al., 2020) or video frame (Wang et al., 2019), respectively.

Our algorithm can also process different levels of granularity, which is important for robust object detection and complex scene description (Wang et al., 2014).

Moreover, snakes allow object recognition with an open-set domain (Kishida et al., 2021), without requiring any cumbersome training.

Besides, snakes can be coupled with ontologies such as the Spatio-Temporal Visual Ontology (STVO) (Olszewska and McCluskey, 2011), and therefore, they can directly bridge vision systems and knowledge-based systems. Furthermore, through STVO, they can be connected to other ontologies which are suitable for cutting-edge vision-embedded technologies such as autonomous systems (Oliveares-Alarcos et al., 2019), cloud-robotic systems (Pignon de Freitas et al., 2020), smart manufacturing (Hildebrandt et al., 2020), or smart cities (Burns et al., 2018).

Hence, all this algorithmic design leads to a transparent and efficient visual object detection and automated semantic scene annotation, while provides an explainable and energy-efficient solution for intelligent vision agents to be deployed in smart cities.

Thus, the contribution of this paper is the new, explainable algorithm that allies recursive, multi-layered snakes with recursive, decision trees for machine-vision object detection and recognition.
2 OUR PROPOSED APPROACH

In this section, we present our ‘Snakes-in-Trees’ (ST) approach, which allows both the automatic visual object detection and recognition as well as its automatic semantic labeling, as exemplified in Fig. 1.

For this purpose, our ST algorithm (Algorithm 1) computes recursively snakes \( S \) in an input image \( I \), in order to detect an object \( O \) in a robust way against occlusions and to map the object’s semantic label \( L \) granularity for a precise object recognition and image annotation.

In this work, each snake \( S \) is implemented by a multi-feature active contour (Olszewska, 2015a) which is defined as a parametric curve \( C(s) : [0,1] \rightarrow \mathbb{R}^2 \) modeled by a B-spline formalism and guided by both the internal forces (\( \alpha \): elasticity, \( \beta \): rigidity) resulting from the curve’s mechanical properties and the external force \( \Xi \) resulting from multiple features of the input image, as per following dynamic equation:

\[
C_i(s,t) = \alpha \cdot C_{ss}(s,t) - \beta \cdot C_{sss}(s,t) + \Xi. \tag{1}
\]

The recursive computation of snakes \( S_{i+1,k} \) within the input image \( I \) is performed by applying \( i+1 \)–times the Eq. (1) to \( I \) and leads to the multi-layered \((l+i)\) partition of \( I \) in terms of object-of-interest’s background (at the layer \( l+i = l \), with \( i = 0 \)), its foreground (at the layer \( l+i = l+1 \), with \( i = 1 \)) as well as the foregrounds of semantically meaningful sub-objects (at the subsequent layers \( l+i \), with \( i \geq 2 \)) of the object of interest delineated by \( k \) snakes at the corresponding layers \( l+i \).

Hence, this recursive process enables the automatic detection of coherent and consistent visual objects, which are described by geometric and metric properties (Olszewska, 2013). These features serve to the characterization of the regions extracted by the snakes and contribute to define the objects or their subparts in terms of both numeric and semantic concepts.

The latter ones are recursively mapped into natural-language keywords through the pre-order traversal of rooted trees that are recursively computed by the ST algorithm, allowing the efficient object labeling as well as accurate object recognition (see Algorithm 1).

It is worth noting that the decision trees, which are recursively built by the ST algorithm, ensure a granular and semantic mapping of the visual objects that are detected in each layer by the snakes, which are themselves recursively computed by the ST algorithm, for an accurate object detection, recognition, and annotation.

Hence, decision trees are induced in order to define semantic keywords at each level corresponding to visual feature defined by each layer of the snakes. Then, a voting mechanism allows for higher semantic level decisions in order to recognize the object.

3 RESULTS AND EVALUATION

To validate our transparent algorithmic method for the automatic visual object detection and recognition as well as automatic image annotation in context of smart cities, we used the publicly available MIT CBCL street scenes database, which contains 35,417 jpg images with a resolution of 1280x960.

We carried out experiments consisting in running our recursive algorithm implemented in Matlab on a commercial device with a processor Intel(R) Core(TM)2 Duo CPU T9300 2.50 GHz, 2 Gb RAM and the Matlab (Mathworks, Inc.) and applied to the CBCL database.
Figure 2: Examples of results of our recursive ‘Snakes-in-Trees’ method, when applied to the CBCL dataset. Best viewed in color.

Table 1: Average accuracy of object-of-interest recognition, using approaches of △(Ren et al., 2015), ○(Lu et al., 2019), ◊(Liu et al., 2008), □(Kim et al., 2004), and our ‘Snakes-in-Trees’ (ST) method.

<table>
<thead>
<tr>
<th>method</th>
<th>△</th>
<th>○</th>
<th>◊</th>
<th>□</th>
<th>ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>average accuracy</td>
<td>65.4%</td>
<td>69.5%</td>
<td>73.1%</td>
<td>84.2%</td>
<td>95.6%</td>
</tr>
</tbody>
</table>

Algorithm 1: ST: Snakes in Trees.

Given $O$, the object of interest to tag;
Given $T_O$, the Tree related to the object $O$;
Given $S_{l+i,k}$, the $k$th Snake at the $l+i$th layer;
Given $L_{O,n}$, the label of the root $n$ of the tree $T_O$;

Considering $T_{O,l+i}$, the $l+i$th level of the tree $T_O$;
i = 0;
$V = \emptyset$;
$r_{S_{l+i,k}} = 0$;

ST($S_{l+i,k}, T_O$)
$L_{O,n} = \text{label(head}(T_{O,l+i}))$
$S_{l+i,k+m} \leftarrow S_{l+i,k}$
$S_{l+i,k} = S_{l+i,k+m}$
if $\neg S_{l+i,k}(S_{l+i,k} = L_{O,n})$ then
    $r_{S_{l+i,k}} = 1$
    $V = V \cup \{r_{S_{l+i,k}}\}$
end if

if $T_{O,l+i,\text{left}} \neq \emptyset$ then
    $i = i + 1$
    ST($S_{l+i,k}, T_{O,l+i,\text{left}}$)
end if

if $T_{O,l+i,\text{right}} \neq \emptyset$ then
    $i = i + 1$
    ST($S_{l+i,k}, T_{O,l+i,\text{right}}$)
end if

if MajorityVote($V$) == 1 then
    return label(head($T_O$))
end if

A sample of obtained results, which have been computed in few ms, can be found in Figs. 2 (a)-(d). We can observe that the algorithm has well detected and recognized the objects of interest as well as their components, delineating and labeling them appropriately.

To quantitatively assess the accuracy of our algorithm, we adopt the standard criterion (Olszewska, 2019b), as follows:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$ (2)

with $TP$, true positive, $TN$, true negative, $FP$, false positive, and $FN$, false negative.

Table 1 reports the mean average accuracy, which is computed as per Eq. (2) in the case of our algorithm (i.e. 95.6%), and compares it to the performance of other state-of-the-art machine-learning methods for object detection and recognition. On one hand, we can note that current methods (Wang et al., 2021) based on deep learning and convolutional neural networks (Fathima and Merriliance, 2020), such as Faster R-CNN (Ren et al., 2015) and Grid R-CNN (Lu et al., 2019), achieve only an accuracy of 65.4% and 69.5%, respectively. As reported by (Khan et al., 2020), CBCL dataset images are challenging because of their low resolution for training as well as testing neural networks (Jimenez-Bravo et al., 2020). On the other hand, common decision-tree-based methods, such as (Liu et al., 2008) and (Kim et al., 2004), have an accuracy of 73.1% and 84.2%, respectively, and they involve constraining assumptions such as recognition of very distant classes only or detection of only centered objects of interest, respectively. Therefore, our ST algorithm features not only explainability, but it is also low-spec, cost-effective, time-efficient, and robust, while being more accurate in comparison with state-of-the-art approaches.
CONCLUSIONS

In this paper, we propose an explainable-by-design algorithm built on snakes within trees for automatic object detection and recognition. Indeed, we have developed an efficient XAI algorithm embedding recursive snakes within the recursive pre-order traversal of rooted trees, where each decision tree’s semantic value has been mapped with the visual information provided by a layer of the computed snake. Hence, based on both semantic and visual properties of the image content, our ‘Snakes-in-Trees’ (ST) method provides accurate and robust object detection and recognition as well as image annotation in real-world conditions, even in case of pose variations or occlusions of the objects of interest. The ST algorithm is thus well suited for smart cities’ intelligent-vision-based applications.

REFERENCES


