Detecting Anomalies in Image Classification by means of Semantic Relationships

Andrea Pasini
DAUIN
Politecnico di Torino
Torino, Italy
andrea.pasini@polito.it

Elena Baralis
DAUIN
Politecnico di Torino
Torino, Italy
elena.baralis@polito.it

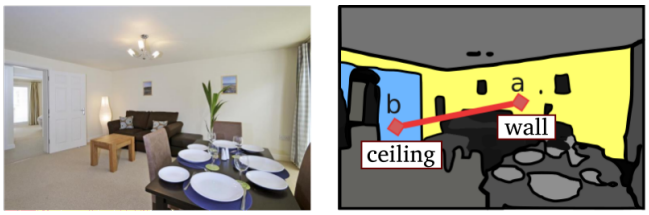
***Abstract*—This paper presents a semantic anomaly detection method (SAD) to detect anomalies in the predictions of any pixelwise semantic segmentation algorithm. This semantic information (e.g., relative positions and sizes of all the object pairs in an image), learned from the training set and stored in a knowledge base as configuration rules, allows the detection of potential misclassifications in the baseline model predictions. Our approach highlights the objects which are not consistent with the contextual information in the knowledge base. It also provides an interpretable motivation for the detected anomaly, based on the semantic information provided by the configuration rules.**

***Keywords*-Semantics; anomaly detection; knowledge discovery; image segmentation**

I. Introduction

Anomaly detection refers to the task of highlighting data which deviate from expected patterns in a dataset. We focused on inspecting anomalies in the results produced by semantic segmentation neural networks [5], [28], [30], whose aim is to assign a class label to each pixel of an image. We look for possible misclassifications by analyzing semantic relationships between the objects detected by the segmentation model. These relationships are derived by considering the contextual information extracted from the whole image, instead of considering separately each object.

Our Semantic Anomaly Detection (SAD) approach generates a set of interpretable rules to decide whether an object is normal or anomalous. In particular, we consider as anomalies the entities which deviate from one or more semantic rules modeling normal data. Consider the anomaly example in Figure 1, detected by SAD in an indoor scenery image taken from the ADE20K [31] dataset (see Section VI). The query image in Figure 1.(a) is segmented and classified by the neural network, producing the result in Figure 1.(b). After this step, objects *a* and *b* have been labeled with the classes *wall* and *ceiling*, respectively. By looking at the relative position between the two objects, our system detects the anomaly reported in Figure 1.(c). The yellow object labeled as *wall* presents a patch of pixels which are directly on the region classified as *ceiling*. By considering the previously extracted rules, SAD detects that the likelihood of finding this configuration is very low (*<0.01*). Indeed, the image classifier erroneously labeled object *b*, whose correct class should be *door* instead of *ceiling*. As shown in this example, the detection of anomalous behaviors in the segmentation results allows the automatic identification of classification errors, enriched by a human understandable description of the anomaly.



a) query image b) segmented image

Anomaly: wall (a) on ceiling (b) has likelihood <0.01

c) model output

Fig. 1: Example of anomaly detected by SAD on the ADE20K dataset.

SAD considers three different types of contextual information: (i) *object co-occurrence* to detect class pairs which often appear in the same image, (ii) *relative position* between objects, and (iii) object *relative size* and space occupancy inside the picture. Our system analyzes a set of training images and automatically builds a *knowledge base*, which describes how the different object classes usually behave for each of these relationship categories. When a new segmented image is provided to SAD, its knowledge base is exploited to detect possible anomalies among the labeled objects. Hence, it is possible to highlight when a pair of objects shows a behavior which deviates from the standard one, modeled by the knowledge base.

Hence, the main contribution of our work is threefold: (i) we propose a novel technique to automatically derive and exploit contextual information represented by means of semantic relationships among objects to detect anomalies, (ii) the proposed approach provides both an interpretable explanation of the detected anomalies and an understandable description of "normality" given by the collection of derived semantic rules, and (iii) the adopted semi-supervised approach does not require ground truth labeling of anomalies, being thus

capable of addressing previously unseen anomaly cases.

A distinguishing feature of SAD is its *interpretability*, given by the possibility to highlight the potentially misclassified objects and the rules in the knowledge base that helped in detecting them. Furthermore, the likelihood of the relationships between the objects may provide a semantic enrichment of the classification result even when the classification is correct.

The paper is organized as follows. Section II provides an overview of related works. Section III introduces the SAD approach, while Section IV covers the definition of the knowledge base. In Section V the anomaly detection algorithm is described, while Section VI evaluates its performance. Section VII draws conclusions and outlines future work.

II. Related Works

Anomaly detection can inspect three types of anomalies: *point*, *group*, and *contextual* anomalies [4]. *Point anomalies* highlight single instances deviating from the common ones in a dataset. *Group anomalies* refer to groups of instances which deviate from the normal behavior. *Contextual anomalies* are defined for instances which are abnormal only in a particular context. Our work focuses on this type of anomalies.

Contextual anomalies are detected in [13], [17] by means of clustering-based frameworks in different application domains. These anomaly detection techniques are unsupervised and based on clustering methods. They differ from our work as they try to group normal data and compute anomalies by looking at data points which are not well described by the generated clusters. Moreover, these methods are not interpretable and do not exploit semantic information to better understand the analyzed data.

Other techniques are based on supervised methods, which need training examples on both anomalies and normal data [2], [6], [29]. However, labeled data is not always available and these methods are not able to detect new types of anomalies which are not in the training data.

Semi-supervised techniques learn by modeling normal instances. Anomalies are detected when some data deviate from the normal behavior. Our method belongs to this category. Typically, semi-supervised techniques are implemented with statistical methods, such as mixture models which learn the probabilistic distribution of normal objects attributes [11], [16]. The SAD approach is different from statistical methods, as it models the behavior of normal data by means of interpretable rules which capture more abstract semantic information with respect to the raw values of the attributes.

In the field of image data, contextual information can be used to discover objects which are unusual with respect to the surrounding ones. In [7] the authors detected irregularities of visual data by analyzing spatial ensembles. Differently from our work, the object positions are defined by means of object centroids and the authors do not use the semantic information of the object classes.

Other works use the contextual information to enhance the performance of image classifiers by considering class co-occurrence [19] and relative positions, which can be modeled either with continuous or discrete values. In [8] the authors used both class co-occurrences and relative positions of the object bounding boxes to enhance a baseline object detector, in which positions are defined as continuous values. Galleguillos *et al.* [12] modeled the relative position between objects with a set of discrete values: *above, below, inside, around*. These values were computed considering the vertical position of the object centroids and the percentage of overlapping of their rectangular bounding boxes. Modeling the position between objects with a discrete set of attributes is more semantically relevant than considering continuous values for the actual coordinates of the objects as in [8]. However, using bounding boxes or centroids [12] to compute the positions is still restrictive, since it does not properly consider objects with more complex shapes as we do in our work.

More recently, reasoning about objects and their properties has been exploited for image captioning and visual question answering (VQA). These tasks are typically addressed by constructing abstract representations of the input scene. The model proposed in [26] infers abstract scene representations with object properties, such as position and pose. However, in this work no object relationships are considered. Other techniques learn abstract concepts by means of generative models [15], [25]. Concepts are defined as set of properties describing an image, such as *smiling face* or *short hair*. The models are trained to generate new images which represent a specific concept. These works consider the image as a whole entity, without focusing on the specific objects as we do in our approach.

VQA is addressed by means of both symbolic reasoning and fully neural network based techniques. Symbolic reasoning can be applied on the object attributes retrieved by Mask R-CNNs to provide answers related to the image [27]. The CNN learns to predict object positions in a supervised way, as it needs for ground truth coordinates. Conversely, in our work we learn relative object positions with an unsupervised approach which does not require additional training information. Fully neural network models substitute the symbolic reasoning task with LSTM neural networks, which directly provide the answers [10]. They avoid the task of designing a reasoning module, but the result is hardly interpretable.

III. The SAD Approach

The Semantic Anomaly Detection process includes several steps, which are depicted in Figure 2 and whose main characteristics are outlined in the following.

Knowledge Base Definition. Our definition of anomaly is based on the availability of a collection of rules, describing the characteristics of "normal" data. SAD learns common patterns in the object configurations for all the different object classes. By analyzing a training set of manually labeled images, SAD builds a knowledge base storing information on mutual object relationships like object co-occurrence, relative position and size. The detected relationships are represented by means of histograms. An in-depth description of properties and techniques to detect them are presented in Section IV.

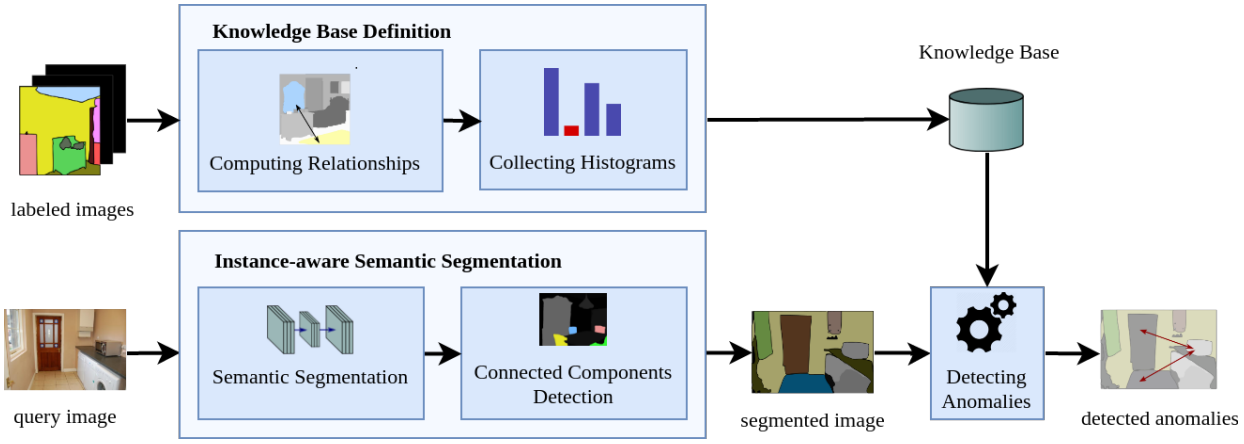


Fig. 2: The SEMANTIC ANOMALY DETECTION process.

Instance-aware Semantic Segmentation. This building block can be partitioned in (i) Semantic image segmentation and (ii) Connected components detection. We addressed the first task by means of *Convolutional neural networks (CNN)*, which are models capable of directly analyzing the pixels of a query image by means of sliding filters, called convolutional layers [5], [20], [28], [30]. For our implementation we chose the *PSPNet* [30] neural network, winner of the ImageNet segmentation challenge in 2016. However, different CNNs [5], [28] can be considered for this task.

Given the neural network output, the Connected Components Detection step locates the object instances in the picture by grouping adjacent pixels which belong to the same class. Each connected component represents an object instance associated to a unique identifier.

Anomaly Detection Given the labeled objects of a query image and the information stored in the knowledge base, this step looks for anomalies in the predictions. The objective is to find object configurations which show a low likelihood according to the information stored in the knowledge base. Section V describes how anomalies are defined and detected, while the effectiveness of the anomaly detection process is validated in Section VI.

IV. KNOWLEDGE BASE DEFINITION

The definition of a semantic rich knowledge base is crucial for the whole anomaly detection process. The SAD knowledge base for image classification includes the following three types of contextual relationships between objects inside the same image: (i) object co-occurrence, (ii) relative object size, and (iii) relative object position.

Before describing object relationships and techniques to extract them from the segmented images, we introduce some definitions.

Definition 1 (Property)**.** *A property p describes a relationship between two objects appearing in the same image. Each property belongs to a particular category c.*

TABLE I: Properties and associated categories.

Category	Properties
position	*above, below, on, hanging, inside, around, side-up, side, side-down*
width	*bigger, same, smaller*
height	*bigger, same, smaller*
area	*bigger, same, smaller*
co-occurrence	*co-occurs*, $\neg$*co-occurs*

Example of categories are the *relative position* or the *relative size*. Table I shows the different properties considered in this work and their associated categories.

Properties and categories allow us to describe the semantic relationships between object classes. We represent these relationships by means of triplets.

Definition 2 (Triplet)**.** *Let s (subject) and r (reference) be two object classes. Let c be the category of a property. A triplet $\mathcal{T} = \langle s, c, r \rangle$ describes the relationships for each property of category c between two objects of class s and r.*

For example, to describe the relative position between *lamp* and *ceiling* we will use $\mathcal{T} = \langle lamp, position, ceiling \rangle$. Histograms are associated to triplets to describe quantitatively the likelihood that the triplet subject and reference satisfy a particular property of a category.

Definition 3 ($\mathcal{T}$-histogram)**.** *Let $\mathcal{T} = \langle s, c, r \rangle$ be a triplet. A $\mathcal{T}$-histogram h($\mathcal{T}$) is given by*

$$h(\mathcal{T}) = [l(p_0), \ldots, l(p_i), \ldots, l(p_{N-1})]$$

where each value $l(p_i)$ specifies the likelihood that the subject s and the reference r of $\mathcal{T}$ satisfy a particular property p_i of category c, and N is the number of properties belonging to category c.

The likelihood is computed from the training pictures as the ratio between the number of images containing two objects belonging to the classes s, r which satisfy property p_i and the total number of images containing the pair s, r. The likelihoods

in the histogram add up to one and can be interpreted as a discrete distribution of probabilities.

For example, the triplet $\mathcal{T}_1 = \langle lamp, position, ceiling \rangle$ and the histogram $h(\mathcal{T}_1)$ = [l(*below*)=0.90, l(*side-down*)=0.1, l(*above*)=0.0, ...] indicate that lamps are 90% of the time below the ceiling. Properties with 0 likelihood in the histogram are omitted for the sake of brevity.

In the following, we define the contextual object relationships and we present techniques to extract them from the segmented images.

A. Co-Occurrence

Objects with similar semantic meaning or functions will appear more likely together in the same image. More specifically, the presence of some particular object categories can be related to the scene type. For example inside an open air picture we would see regions like *sky, sun, street*, but not others such as *cabinet* or *ceiling*. Hence, the co-occurrence probability models the relationship between different types of objects appearing together in the same pictures.

We model the co-occurrence probability by analyzing the training set images. Similarly to frequent itemsets mining [3], we consider each picture as a transaction and the set of contained object classes as its items. If an image contains more than one instance of the same object class (i.e., item), the item is considered once. To describe co-occurence we use the *Certainty Factor (CF)* measure, which was defined in the expert system MYCin [14], [23]. It allows detecting whether the presence of objects of class r (reference) increases or decreases the probability of having also (objects of class) s (subject) in the same transaction. It is defined as the difference between the Measure of Belief (MB), which specifies the increase of probability of having s in a transaction containing r, and the Measure of Disbelief (MD), which specifies the decrease of probability of s given r.

The *CF* is defined as [23]:

$$CF(s,r) = \begin{cases} \frac{P(s|r)-P(s)}{1-P(s)}, & \text{if } P(s|r) > P(s) \\ \\ \frac{P(s|r)-P(s)}{P(s)}, & \text{otherwise} \end{cases}$$

The value of CF ranges between -1 and $+1$. When it is greater than zero the presence of r encourages the presence of s. Otherwise, if it is less than zero, the presence of s in the transaction is discouraged because of r.

Compared to lift [18], the CF provides more actionable information because it is not symmetric for the two classes. Furthermore its value is constrained in a fixed, more manageable, range. Confidence [18], instead, does not relate the presence of one class to the absence of the other.

Class co-occurrences are modeled in the knowledge base as triplets $\mathcal{T} = \langle s, \textit{co-occurrence}, r \rangle$ (where s and r are classes) with the following histogram:

$$\begin{aligned} h(\mathcal{T}) = & [l(\textit{co-occurs}) = CF_{norm}, \\ & l(\neg \textit{co-occurs}) = 1 - CF_{norm}] \end{aligned}$$

where CF_{norm} is the certainty factor normalized between 0 and 1, i.e., $CF_{norm} = (CF(s,r)+1)/2$

B. Object Sizes

This type of contextual information compares the relative size between objects in the same image. The relative space occupancy is computed in three different ways: *width, height* and *area*.

Width, Height. Considering the bounding box related to an object, we compute its width and height in percentage with respect to the picture size. In this way we compare the bounding boxes independently of the image size and separately for width and height. Analyzing separately the two dimensions allows us to characterize objects which are taller than others (e.g., a door with respect to a table), or wider (e.g., the ceiling or the sky with respect to a tree). The relative size is described by three different properties: *bigger, same, smaller* (see Table I).

Let s denote the subject and r the reference for the category *width*. We define:

$$width(s,r) = \begin{cases} \textit{bigger}, & \text{if } w(s)/w(r) > 1 + thr \\ \textit{smaller}, & \text{if } w(r)/w(s) > 1 + thr \\ \textit{same}, & \text{otherwise} \end{cases}$$

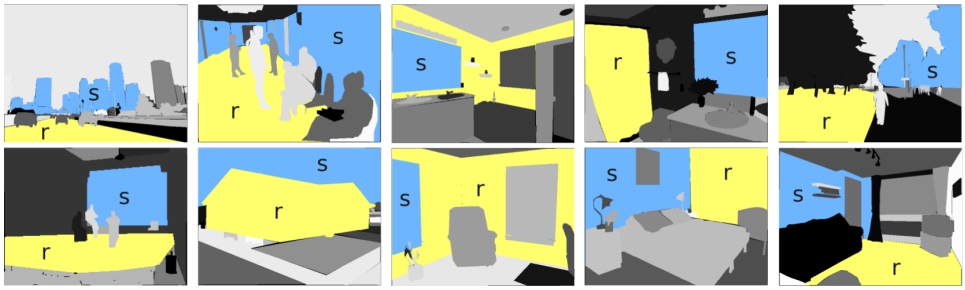
where $w(s)$ and $w(r)$ are the width of the bounding boxes of s and r, and thr specifies the percentage of tolerance to establish the relationship.

For example, in the comparison *"A is 70% bigger than B"* r is *A*, s is *B* and the relationship is verified with $thr >= 0.7$. Experimental evaluation (not reported in this paper) allowed us to detect that *thr* values may range between 0.7 and 0.9 without affecting the results of the anomaly detection phase. Hence, for our experiments we set thr to 0.8. Relative height is computed similarly.

Area. Some objects may occupy very large regions in term of bounding boxes, but their pixels cover only a small area inside the rectangle. The actual area of the object is a measure of its visual weight inside the image, which does not depend on the bounding box width and height. For example objects with holes, such as a *ladder*, or a *fence* are characterized by a small area even if their bounding box covers a big region. The area of an object is computed by considering the number of pixels of its associated connected component. The comparison between a subject and a reference is computed as shown for the properties *width* and *height*.

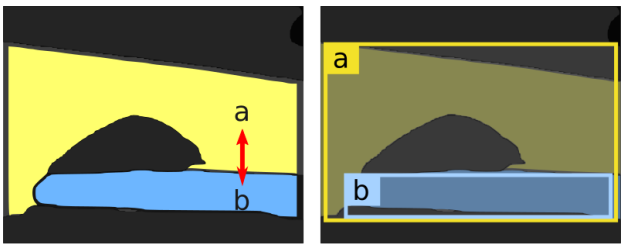
C. Object Positions

The relative position between objects is relevant for scene understanding. It might be easily computed by looking at the bounding boxes of the objects. However, its correct detection may be difficult. Consider for instance Fig. 4 which shows an example of outdoor scenery. On the left (Fig. 4.(a)), *object a* (a bridge) is visually *on object b* (a river). However, if we only consider the bounding boxes, the correct relationship cannot be inferred, as shown in Fig. 4.(b) in which the two rectangles are one inside the other. These issues typically occur with



a) above b) on c) inside d) side e) side up

Fig. 3: Object positions. The images show the relationships between subject (light blue region marked with s) and reference (yellow region marked with r) objects.



a) actual shape b) bounding boxes

Fig. 4: Actual shape and bounding boxes.

objects whose shape is irregular and cannot be approximated with a simple rectangle. Some works show how to infer the object positions from bounding boxes [21] or point clouds [22]. However, these methods do not consider the real shape of the objects or specify only few types of descriptors for the relative position. As discussed in the following, we extract a variety of different positional relationships between objects.

We model the relative position of objects with a variety of properties, listed in Table I, which we describe in the following. Some of them are exemplified in Figure 3, in which s (subject) and r (reference) denote the two terms of paragon.

The *above/below* properties relate a subject which is directly on the same horizontal position of the reference, but it is separated vertically by some interleaving region. Figure 3.(a) shows two examples of the *above* property. In the first one a *building* is above the *street*, while the second one represents a room where a *slide viewer* is above the *floor*, separated by a thin region belonging to the wall.

When the two objects are horizontally aligned and touching, we define the *on* property if the subject is at the top of the reference and *hanging* when the subject is below the reference (Figure 3.(b)).

When the two objects are one inside the other, we define the properties *inside* (subject inside the reference), as in Figure 3.(c), and *around* (reference inside the subject).

Finally, the *side* property refers to a pair of objects which are aligned vertically and occupy different horizontal positions (Figure 3.(d)). In addition we use *side-up* (Figure 3.(e)) and *side-down* to specify when the subject is neither aligned horizontally, nor vertically with the reference.

The presented properties are mutually exclusive (only one can be assigned to an object pair) and symmetric. For example if the subject is *above* the reference, then the reference is *below* the subject. Properties are computed by analyzing object pairs in the image by means of a variant of the string representation proposed in [24].

D. Knowledge Base Definition

Knowledge is stored in the knowledge base in the form of configuration rules which are used to model the statistical information of the object relationships for each class pair. Each rule is composed of a *triplet* and a *histogram*. The knowledge base is built by iterating over the training images and collecting in the histograms the frequency of the different object configurations.

After this process we apply two filtering operations to keep only the most relevant information in the knowledge base. The first filter models the concepts of always/never, while the second selects the most reliable histograms whose values are learned by a minimum number of training samples.

Relevant histograms present an unbalanced distribution of the likelihoods and may specify concepts such as "*ceiling is never below floor*" or "*chair is always on the floor*". For example with the histogram *[l(above)=0.99, l(below)=0.0, l(inside)=0.01, ...]* we model a situation where the subject is (almost) always above the reference. Conversely, the histogram *[l(above)=0.7, l(below)=0.0, ...]* models the concept "subject never below reference". The first filter only selects histograms h which satisfy at least one of the following constraints:

- h contains one likelihood $l(p_i) > thr_h$
- h contains at least one likelihood $l(p_i) < 1 - thr_h$

where thr_h is a threshold whose value will be discussed in Section VI. The first constraint selects histograms presenting a very high likelihood value (i.e., modeling the *always* concept). The second one selects histograms which model the concept *never*, as they show one or more very low likelihoods.

The second filter selects among the remaining histograms only the ones with a minimum frequency, i.e., a minimum support $minsup$, whose value is discussed in Section VI. The support is defined as the number of object pairs in the training set used to collect the statistics for the histogram.

V. Anomaly Detection

Anomalies are defined as object pairs which do not satisfy one or more configuration rules in the knowledge base. Configuration rules allow the detection of object configurations which deviate from the behavior of normal instances.

The procedure for detecting anomalies is shown in Algorithm 1. Let $\mathcal{I}$, *KB* be a segmented image and the knowledge base, respectively. The process iterates (lines 2-3) over all the object pairs S (subject), R (reference) in $\mathcal{I}$ and the categories c defined in Section IV. At each iteration we compute the property p of category c between S and R (line 4), as described in Section IV. Afterwards, we extract the corresponding configuration rule from the knowledge base by reading its triplet $\mathcal{T}$ with category c and the class labels of S, R and its histogram $h(\mathcal{T})$ (lines 5-6). From $h(\mathcal{T})$ the likelihood $l(p|\mathcal{T})$ that two generic objects with the same classes of S, R satisfy p is computed. Similarly, the likelihood that p is not satisfied is computed as the complement of $l(p|\mathcal{T})$ (lines 7-8).

When the likelihood $l(\neg p|T)$ is higher than threshold thr_h, defined in Section IV-D, an anomaly is detected with confidence $conf = l(\neg p|\mathcal{T})$ (lines 9-10). Anomalies have stronger confidence when the likelihood of p is lower. The detected anomalies are stored in the set $\mathcal{A}n$.

After detecting anomalies, the SAD approach exploits them to label each object as *normal* or *exception* with respect to its contextual information. This task can be interpreted as a binary classification task, which labels objects as *normal* or *exception*. Our method assigns the exception class to all the objects which are either the subject or the reference of at least one contextual anomaly in $\mathcal{A}n$.

Algorithm 1 SAD: anomaly detection

Input: Segmented image $\mathcal{I}$, Knowledge Base KB
Output: $\mathcal{A}n$

1: $\mathcal{A}n$ = {}
2: **for all** (S, R) in objectPairs($\mathcal{I}$) **do**
3: **for all** c in *categories* **do**
4: p = computeProperty(S, c, R)
5: $\mathcal{T}$ = $\langle label(S), c, label(R)\rangle$
6: $h(\mathcal{T})$ = getHistogram(KB, $\mathcal{T}$)
7: $l(p|\mathcal{T})$ = getLikelihood($h(\mathcal{T})$, p)
8: $l(\neg p|\mathcal{T})$ = 1 - $l(p|\mathcal{T})$
9: **if** $l(\neg p|\mathcal{T}) > thr_h$ **then**
10: $\mathcal{A}n$ = $\mathcal{A}n \cup$ anomaly(S, R, $conf = l(\neg p|\mathcal{T})$)
11: **end if**
12: **end for**
13: **end for**
14: **return** $\mathcal{A}n$

VI. Experimental Results

The experiments to evaluate our approach are performed on the MIT Scene Parsing Benchmark [1], whose data comes from the *ADE20K dataset* [31]. The benchmark consists of a big collection of *20,000* training images and *2,000* test samples, related to both indoor and outdoor sceneries. Each image is pixel-wise annotated with objects from 150 different categories. A small percentage of the pixels is not labeled and is ignored by our algorithm, as specified by the challenges on this benchmark. SAD builds the knowledge base by inspecting the ground-truth labels of the training images, while it applies anomaly detection to the test images, labeled with the PSPNet model [30]. For reproducibility purposes, SAD source code is available at the following link: https://github.com/AndreaPasini/SAD2019.

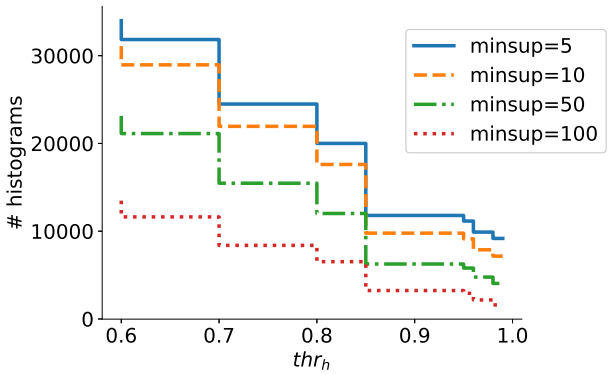


Fig. 5: Histograms for different *minsup* and thr_h values.

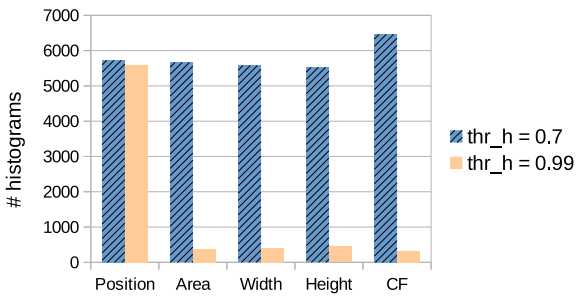


Fig. 6: Histograms for each category. $minsup$=10.

Before discussing anomaly detection results we analyze the statistics collected in the knowledge base. Figure 5 shows the number of relevant histograms while varying the minimum support ($minsup$) and the threshold thr_h (see Section IV-D). On the considered dataset, we obtain up to 34,000 histograms for $minsup = 5, thr_h = 0.6$ and about 13,000 for $minsup = 100, thr_h = 0.6$. Figure 6 shows the number of histograms for each category with two different values of thr_h. Note that if $thr_h = 0.7$ the different histogram types are balanced in number. When thr_h is increased to 0.99, the relevant histograms for the position are more numerous than the others. Hence, the relative position may provide more reliable information than the other categories.

Table II shows, for a selection of class pairs *s*-*r*, the corresponding certainty factors $CF(s, r)$ modeling co-occurrence. In the left section of the table we can observe examples of positive CF, which entail that the presence of class *r* enhances the probability of finding an object of class *s* in

TABLE II: Certainty Factor examples.

Class Pair	CF	Class Pair	CF
wall, oven	1.00	sky, microwave	-0.99
wall, sink	0.99	cabinet, road	-0.99
floor, sofa	0.96	sofa, car	-0.99
bed, pillow	0.94	sky, countertop	-0.99
building, sidewalk	0.93	floor, hill	-0.98
sky, mountain	0.91	lamp, river	-0.98

TABLE III: Area relationship examples.

Class Pair	Sup	Histogram
plate, swivel chair	25	bi=0.00 sa=0.00 sm=1.00
light, microwave	378	bi=0.02 sa=0.06 sm=0.92
runway, van	20	bi=0.95 sa=0.05 sm=0.00
painting, pool table	271	bi=0,03 sa=0,04 sm=0,94

the same image. Considering for example the pair *wall-oven*, the CF is 1.0 and indicates that *ovens* are probably found into a kitchen scenery which most likely contains a *wall*. Observe that even if the $CF(wall, oven)$ is very high, the symmetric $CF(oven, wall)$ only takes the value 0.008. Hence, the symmetric reasoning is not valid, since the presence of a *wall* does not encourage the probability of finding an *oven*. Indeed, many indoor scenes present a *wall* object, but only few of them also the class *oven*. The $CF(oven, wall)$ is however still positive, which means that the presence of *wall* does not influence negatively the one of *oven*. The right column of the table presents some examples of negative CF. A negative value specifies that the presence of the second class (for example *microwave*) inhibits the probability of the first one, for example *sky*. In the provided samples we can notice pairs mostly containing objects which belong to different environments, such as indoor/outdoor scenes.

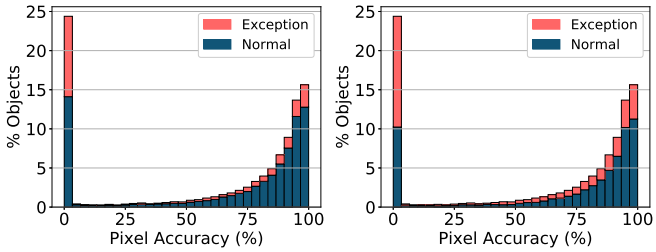
Table III reports some area relationships between different object classes. The first column specifies the class pair (subject-reference), while the second one presents the support of the collected statistics. The third column finally shows the histogram distribution for the three properties (*bigger*/*same*/*smaller*). For example, 95% of the time runways are bigger than vans and 94% of the times a *painting* is smaller than a *pool table*. Finally, Table IV shows some examples of relative positions between objects. To simplify visualization we only present the most relevant likelihood values of each histogram. For example, the first one specifies that *runways* are mostly below *sky* and the second one describes how an object *ball* is mostly inside *pool tables*.

A. Anomaly Detection on ADE20K Dataset

We model anomaly detection as the binary classification task of labeling objects as *normal* or *exception*. We expect that the objects labeled as exception may be misclassified by the semantic segmentation process. In particular, they should have a low pixel accuracy. Pixel accuracy, defined as the ratio between the number of pixels with the correct class over the total number of pixels, is a standard metric to evaluate segmented images [9].

TABLE IV: Position relationship examples.

Class Pair	Sup	Histogram
runway, sky	151	*below*=0.87 *side-down*=0.1
ball, pool table	33	*inside*=0.91 *above*=0.03
light, sink	1321	*side-up*=0.83 *above*=0.17
armchair, cradle	35	*side*=0.8 *side-up*=0.06
painting, pillow	1709	*side-up*=0.6 *above*=0.3 *side*=0.1
bus, path	31	*side-up*=0.6 *above*=0.16 *on*=0.1
curtain, window	8077	*side*=0.6 *on*=0.14



(a) Experiment a (b) Experiment b

Fig. 7: Objects labeled as *normal* or *exception*.

Based on the intuition that objects with lower pixel accuracy more likely present contextual anomalies, we define true and false positives in the following way. An object labeled as *exception* is a true positive if its pixel accuracy obtained by the semantic segmentation model is lower than 75%. Objects with higher accuracy and the *exception* label are considered false positives. In the following we describe the results obtained on the 2,000 test images labeled with PSPNet [30]. We define two experiment configurations: *Experiment a* (Figure 7a) and *Experiment b* (Figure 7b). For *Experiment a* we set $minsup = 10$, $thr_h = 0.98$ for co-occurrence, $thr_h = 0.99$ for position and size. In *Experiment b* we set $minsup = 10$, $thr_h = 0.98$ for co-occurrence, $thr_h = 0.97$ for position and size. The latter configuration with a relaxed constraint on position/size histograms allows inspecting a trade-off with higher recall with respect to *Experiment a*.

Table V shows for each category the number of anomalies detected in the test images. Both the total and percentage of anomalous object pairs are shown. The categories with the largest number of detected anomalies are *position* and *co-occurrence*, with values from 1242 to 3347.

Figures 7a, 7b depict the result of the anomaly detection phase. Each bin of the histogram charts shows the number of objects with a specific pixel accuracy. The number of objects is specified in percents with respect to the total in the test set. For example we can observe that the semantic image segmentation step classified almost 25% of the objects with a very low accuracy. The upper part of the bins (in red) specifies the percentage of objects which are labeled as *exception*. *Experiment b*, with a more relaxed parameter setting, yielded more detections than *a*. From the charts it can be also noticed that anomalies do not only cover objects with low accuracies.

The analysis shows that the SAD approach can detect very well many low accuracy objects. However, distinguishing the

TABLE V: Number of detected anomalies in 2000 test images.

Experiment	Total	Percent (%)	Position	Area	Width	Height	CF
experiment a (Fig. 7a)	4581	6.4	1242	35	14	20	3270
experiment b (Fig. 7b)	6731	9.3	3347	69	19	26	3270

TABLE VI: Precision and recall for the *exception* and *normal* classes.

Experiment	Precision (Ex)	Recall (Ex)	Precision (Norm)	Recall (Norm)
experiment a	0.6536	0.3601	0.5996	0.8339
experiment b	0.6152	0.5230	0.6328	0.7153

(reduced number of) false positives (i.e., objects with high pixel accuracy covered by many anomalies) is still challenging.

VII. Conclusions and Future Work

In this paper we presented SAD, a novel method for semantic anomaly detection in segmented images. SAD extracts from a set of labeled images a collection of interpretable semantic configuration rules stored in a knowledge base, which is the core of our approach. SAD detects contextual anomalies, defined as the objects which do not conform to normal instances modeled by the knowledge base rules. The experiments performed on the ADE20K dataset confirmed that the detected anomalous objects are potential misclassifications, as they are characterized by a low pixel accuracy.

As future work we will extend the available semantic information by using prior knowledge extracted from ontologies to model more complex semantic relationships between objects, such as their usage and functionalities. A further possible application of our model is to consider the presence of anomalies to evaluate the quality of segmented images when the ground truth labels are not available.

References

[1] MIT scene parsing benchmark, http://sceneparsing.csail.mit.edu/. 2016.

[2] T. Abbes, A. Bouhoula, and M. Rusinowitch. Efficient decision tree for protocol analysis in intrusion detection. *International Journal of Security and Networks*, 5(4):220–235, 2010.

[3] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.

[4] M. Ahmed, A. N. Mahmood, and J. Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.

[5] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE trans. on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[6] D. Barbará, J. Couto, S. Jajodia, and N. Wu. ADAM: a testbed for exploring the use of data mining in intrusion detection. *ACM Sigmod Record*, 30(4):15–24, 2001.

[7] O. Boiman and M. Irani. Detecting irregularities in images and in video. *International journal of computer vision*, 74(1):17–31, 2007.

[8] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchical context on a large database of object categories. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conf. on*.

[9] G. Csurka, D. Larlus, F. Perronnin, and F. Meylan. What is a good evaluation measure for semantic segmentation?. In *BMVC*, volume 27, page 2013. Citeseer, 2013.

[10] M. T. Desta, L. Chen, and T. Kornuta. Object-based reasoning in VQA. *arXiv preprint arXiv:1801.09718*, 2018.

[11] E. Eskin. Anomaly detection over noisy data using learned probability distributions. In *Proc. of the Int. Conf. on Machine Learning*. Citeseer, 2000.

[12] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *Computer Vision and Pattern Recognition (CVPR), 2008. IEEE Conf. on*, pages 1–8. IEEE.

[13] M. A. Hayes and M. A. Capretz. Contextual anomaly detection in big sensor data. In *Big Data (BigData Congress), 2014 IEEE Int. Congress on*, pages 64–71. IEEE, 2014.

[14] D. Heckerman. The certainty-factor model. *Encyclopedia of Artificial Intelligence,*, pages 131–138, 1992.

[15] I. Higgins, N. Sonnerat, L. Matthey, A. Pal, C. P. Burgess, M. Bošnjak, M. Shanahan, M. Botvinick, D. Hassabis, and A. Lerchner. SCAN: Learning hierarchical compositional visual concepts. 2018.

[16] R. Laxhammar, G. Falkman, and E. Sviestins. Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator. In *Information Fusion, 2009. FUSION'09. 12th Int. Conf. on*, pages 756–763. IEEE, 2009.

[17] Q. Liu, R. Klucik, C. Chen, G. Grant, D. Gallaher, Q. Lv, and L. Shang. Unsupervised detection of contextual anomaly in remotely sensed data. *Remote Sensing of Environment*, 202:75–87, 2017.

[18] A. K. V. K. Pang-Ning Tan, Michael Steinbach. Introduction to data mining, 2nd edition, 2018.

[19] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Computer vision, 2007. ICCV 2007. IEEE 11th int. conf. on*, pages 1–8. IEEE, 2007.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[21] W. Ren, M. Singh, and S. Singh. Image retrieval using spatial context. In *Proceedings of the 9th international workshop on systems, signals and image processing*, pages 44–49, 2002.

[22] B. Rosman and S. Ramamoorthy. Learning spatial relationships between objects. *The Int. Journal of Robotics Research*, 30(11):1328–1342, 2011.

[23] E. H. Shortliffe and B. G. Buchanan. A model of inexact reasoning in medicine. *Mathematical biosciences*, 23(3-4):351–379, 1975.

[24] J. R. Smith et al. Decoding image semantics using composite region templates. In *Content-Based Access of Image and Video Libraries, 1998. Proceedings. IEEE Workshop on*, pages 9–13. IEEE, 1998.

[25] R. Vedantam, I. Fischer, J. Huang, and K. Murphy. Generative models of visually grounded imagination. *arXiv preprint arXiv:1705.10762*, 2017.

[26] J. Wu, J. B. Tenenbaum, and P. Kohli. Neural scene de-rendering. In *Proc. CVPR*, volume 2, 2017.

[27] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum. Neural-symbolic VQA: Disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems*, pages 1039–1050, 2018.

[28] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[29] Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, and J. Ucles. Hide: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In *Proc. IEEE Workshop on Information Assurance and Security*, pages 85–90, 2001.

[30] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, 2016.

[31] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20k dataset. In *Proc. CVPR*, 2017.

Acknowledgment

This work has been supported by the SmartData@PoliTO center on Big Data and Data Science.