ODROM: Object Detection and Recognition supported by Ontologies and applied to Museums

Alejandro Tejada-Mesias, Irvin Dongo, Yudith Cardinale, Jose Diaz-Amado

ODROM: Object Detection and Recognition supported by Ontologies and applied to Museums

Alejandro Tejada-Mesias
Electrical and Electronics Engineering Department
Universidad Católica San Pablo
Arequipa, Peru
alejandro.tejada@ucsp.edu.pe

Irvin Dongo
Electrical and Electronics Engineering Department
Universidad Católica San Pablo
Arequipa, Peru
Univ. Bordeaux, ESTIA INSTITUTE OF TECHNOLOGY
Bidart, France
ifdongo@ucsp.edu.pe

Yudith Cardinale
Electrical and Electronics Engineering Department
Universidad Católica San Pablo
Arequipa, Peru
Universidad Simón Bolívar
Caracas, Venezuela
ycardinale@usb.ve

Jose Diaz-Amado
Electrical and Electronics Engineering Department
Universidad Católica San Pablo
Arequipa, Peru
Instituto Federal da Bahia
Vitoria de Conquista, Brazil
jose_diaz@ifba.edu.br

***Abstract*—In robotics, object detection in images or videos, obtained in real-time from sensors of robots can be used to support the implementation of service robot tasks (e.g., navigation, model its social behavior, recognize objects in a specific domain), usually accomplished in indoor environments. However, traditional deep learning based object detection techniques present limitations in such indoor environments, specifically related to the detection of small objects and the management of high density of multiple objects. Coupled with these limitations, for specific domains (e.g., hospitals, museums), it is important that the robot, apart from detecting objects, extracts and knows information of the targeted objects. Ontologies, as a part of the Semantic Web, are presented as a feasible option to formally represent the information related to the objects of a particular domain. In this context, this work proposes an object detection and recognition process based on a Deep Learning algorithm, object descriptors, and an ontology. ODROM, an *O*bject *D*etection and *R*ecognition algorithm supported by *O*ntologies and applied to *M*useums, is an implementation to validate the proposal.**

Experiments show that the usage of ontologies is a good way of desambiguating the detection, obtained with a mAP@0.5=0.88 and a mAP@[0.5:0.95]=61%.

***Index Terms*—Deep Learning, object detection, Ontology, museums, service robots**

I. Introduction

[1]

Nowadays, object detection is a widely used technique that allows identifying objects present in images or videos, obtained from numerous sources, such as robot sensors, security cameras, or mobile devices [36]. The areas that have benefited from the advancement and development of object detection are numerous and include autonomous driving [6], [18], face detection [28], pedestrian detection [35], robotics and service robots [4], [24], [32]. Particularly in robotics, object detection in images and videos obtained in real time through the sensors of robots, can support the implementation of tasks of service robots, such as navigation, modeling of social behavior, object recognition of a specific domain (e.g., artworks, kitchen utensils, elements in a restaurant), which are usually done in (*indoor environments)* [2], [7], [15].

However, the object detection in indoor environments, based on traditional *Deep Learning* techniques, have limitations when there is a high density of objects in small spaces and when the objects are small [5]. Thus, new learning models are required to overcome these limitations.

In the context of service robots, there are several applications (e.g., hospitals [4], restaurants [24]) in which only the use of object detection is not enough to develop the tasks satisfactorily and therefore, it is necessary to search and implement additional solutions that allow obtaining intrinsic information from objects (e.g., color, shape, semantics) and thus, provide greater robustness to the information handled by robots [8]. Therefore, there is a need to manage this large amount of information regarding objects, which in turn implies the generation of complex knowledge. To better manage this information, i.e., this type of knowledge, it is necessary to have a formal and standardized representation. Ontologies, as part of the Semantic Web, are presented as a feasible and attractive option to model such knowledge, offering a standard representation of construction and application, as well as great flexibility[2].

[2]https://www.w3.org/standards/semanticweb/ontology

In this context, this work proposes an object detection technique based on: (i) *CNN (Convolutional Neural Networks)*, capable of operating in indoor environments with high density of objects; (ii) an intrinsic object disambiguation technique, based on additional characteristics of the detected object such as the dominant color; and (iii) a semantic repository, based on ontologies, which offers an enriched source of information regarding the objects detected. To demonstrate its functionality and performance, the proposal is implemented within the context of the RUTAS *(Robots for Urban Tourism centers, Autonomous and Semantic based)* Project, whose objective is to develop service robots that works as urban tourism guides in indoor environments, such as museums, in order to arouse interest in culture and, in turn, preserve the cultural and historical heritage through the application of technology. This implementation is called ODROM (**O***bject* **D***etection and* **R***ecognition supported by* **O***ntologies and applied to* **M***useums*). ODROM receives information through the sensors of the service robots that capture images and videos, detecting the artworks during their journey. The network is trained using a dataset with photos of artworks from two museums in Arequipa, Peru (La Recoleta Museum and Municipal Museum of Arequipa) and the ontology of museums CURIOCITY (Cultural Heritage for Urban Tourism in Indoor/Outdoor environments of the CITY) [21], which contains detailed information on those artworks.

Experiments show that the training carried out with the specialized museum-dataset achieved adequate results (mAP@0.5=88% and mAP@0.5:0.95=61%); and the use of ontologies combined with the object detector is an effective strategy of disambiguation of the object classification.

This paper is organized as follows. In Section II, related concepts to understand this work are described. Section III describes and classifies the related work. In Section IV, the different stages of the proposal are presented. Section V presents an implementation of our proposal called ODROM. The results obtained in the training and in the interaction of the CNN algorithm with the ontologies are reported. Additionally, Section VI provides general comments, important insights, and challenges to be solved. Finally, in Section VII the conclusions and future work are described.

II. Preliminaries

In this section, the most relevant concepts related to *Deep Learning* and *Semantic Web* areas are described in order to understand the proposal.

A. Algorithms for Object Recognition

To carry out object recognition in any application, it is common to use algorithms based on Deep Learning, capable of extracting and learning characteristics of the objects that are used during training and validation, to later be applied as detectors in the application of interest [36]. These algorithms mostly divide the detection process into three stages:

- Detection Stage: In this first stage, the characteristics learned in the training are used to detect the different objects that may be present in an image. The best options to develop this stage are the *CNN (Convolutional Neural Networks)* [16]. These make use of the convolution as a mathematical tool to obtain the characteristics of the objects present in the images, as well as to use them in detection. To perform this task, the image is scanned by pixels with a kernel that can vary in size depending on the network configuration (e.g., Alexnet [16] uses 5x5 or 7x7 kernels, while *VGG (Visual Geometry Group)* [23] uses a 3x3); the end product of this scan is a *feature map*. To speed up the process, a size reduction of these maps is performed, known as *Pooling*. This task can be *MaxPooling* (selection of the highest value within the scan window) or *Average Pooling* (which averages the values within the scan window). The more extraction and *Pooling* layers there are, the more generalization of data characteristics exists, but information related to details is also lost. To avoid the presence of negative numbers in feature maps, *ReLU (Rectified Linear Unit)* layers are used, which convert negative values to zero and reduce the complexity of the system. This layer is located before the *Pooling* layers.
- Localization Stage: The objective of this second stage is to mark the objects detected in the previous stage with a *Bounding Box*. To do so, the algorithm encloses the space where the detected features have a high degree of similarity, i.e., the space where there is an object. On many occasions, the algorithm tries several times to enclose the object in question, being able to generate various predictions and therefore not a specific one. For this, *NMS (Non-Maximum Suppression*) is applied, which uses *IoU (Intersection Over Unit* as a minimum threshold and allows obtaining the best of predictions. During training, the object in question is enclosed within a *Ground Truth Bounding Box*, to make the algorithm extract the characteristics directly from what is within it.
- Classification Stage: It is the final stage that seeks to assign a class to the detected and located object. To do this, a comparison is made of the characteristics extracted from the object with the saved characteristics belonging to each class. The class that is most similar to the characteristics of the object is assigned.

Even if several advances in the area of object detection for images and videos have been performed, the most widely used algorithms still have difficulties [36]:

- Intraclass variations: These are the differences that two objects which belong to the same class can present due to differences in models or shapes.
- Different points of view and lighting: The same object can be observed from different angles and heights, as well as under different light sources or times of the day.
- Object rotation: It is likely that something or someone rotates the object at one time to another and this can alter the results of the detection at different times.
- Scale changes: If the object is zoomed in or out, the object

changes its size and, therefore, this can generate errors.
- Detection of objects in density or occluded: The first refers to detection in images with a high number of objects, while the second one refers to developing the task with the object of interest partially obstructed by another one.
- Acceleration of detection algorithms: It refers to the speed to perform the recognition. In real-time or near-real-time scenarios, this task needs to run faster and faster.

B. Semantic Web

The Semantic Web, according to the information from the *World Wide Web Consortium (W3C)*, is an extension of the current Web and aims to provide to applications information with understandable and interpretable metadata, in order that computers can establish relationships automatically. To achieve this, several Web technologies are used, such as: *IRI (International Resource Identifier)*, the evolution of *URI (Uniform Resource Identifier)*, *XML (eXtensible Markup Language)*, language used by the Semantic Web to encode the data through a semi-structured language, *RDF (Resource Description Framework)*, standard approved by the W3C that allows representing any type of resources on the Web, identifying and ordering them; and the *OWL (Ontology Web Language)* which is a standard language created for the more complex knowledge of objects and their relationships, defining classes, subclasses and properties in order to express an effective domain.

The ontologies are composed of three types of metadata or representational primitives:

- Classes (or *sets*): They represent the objects and categories to which the different entities of the domain of knowledge represent. The object detection stage can be matched with these classes of the ontology.
- Relationships: They represent an analogous version to the relationships that different objects have with each other in real life. They can be determined by phrases or verbs.
- Attributes (or Properties): These are the characteristics of the classes or objects, which represent important and intrinsic data to themselves.

To access the information stored in semantic repositories, it is necessary to use a query language different from that of traditional database queries (SQL). Thus, to access the RDF data, the SPARQL query language is used. This has four different types of requests and each one is oriented to a specific function:

- `Ask`: Its objective is to obtain as a response if there is at least one value within the set that is equivalent to some value of the resource.
- `Select`: It is used to select a portion or all of the collated data in table form (including sampling, paging and aggregation through an *offset* as the initial value and a *limit* as the final value).
- `Construct`: Build an RDF graph with all variables collated together.
- `Describe`: It provides descriptions of the collated data against the construction of a relevant RDF graph.

Following section describes the related work.

III. Related Work

Based on the limitations of object detection models (see Section II-A), the most recent related works are focused on overcoming these difficulties. We describe the most relevant below.

A. Object Detectors

Over the years, several researchers have developed specialized algorithms for this application. The most efficient and therefore best used are classified into two groups:

- Two-stage detectors: These are algorithms that appeared at the beginning of 2014 and are characterized by having two well-marked stages: one to find *ROI (Region of Interest* and snip them out, and the other to classify them. The most recognized detectors for this task are:
  - Regions CNN (RCNN) [12]: Appeared in 2014 and makes use of Selective Search. Images are scaled to a predetermined size and passed through a pre-trained CNN, and then classified through a *SVM (Support Vector Machine*. It has a *Mean Average Precision* (mAP) of 58.5% and takes 14 seconds per image
  - Fast RCNN [11]: Developed in 2015, it has the advantage that the training of *Bounding Boxes* and the detector are performed under the same network configurations. It has an efficiency of mAP = 70.0% (with the VOC07 dataset [3]
  - Faster RCNN [27]: Introduced in 2015, it was the first end-to-end detector and the first to make use of *Deep Learning*, with a speed close to real time. It had efficiencies of mAP = 73.2% (with the VOC07 *dataset*), mAP = 42.7% and mAP @ [0.5,0.95] = 21.9% (with the COCO *dataset* [4]). It introduced the *RPN (Region Proposal Networks)*, networks that provide location proposals almost without adding computational cost. Its main disadvantage is the redundancy in computations in the subsequent detection stages.
- CNN-based one-stage detectors: These algorithms have, as their main goal, to offer a speed improvement while maintaining efficiency as much as possible. They eliminated the snipping and scaling stage to have only one to do those functions. The most used algorithms in this category are:
  - You Only Look Once (YOLO) [26]: Its first version appeared in 2015 and its main feature was a speed of 155fps *(Frames Per Second)* Its efficiency is mAP = 52.7% (with the VOC07 *dataset*) and its main

[3]http://host.robots.ox.ac.uk/pascal/VOC/voc2007/
[4]https://cocodataset.org/#home

disadvantage is the difficulty to recognize small objects. The network has been updated over the years, currently having version 5, developed by *Ultralytics*.

– RetinaNet [19]: Appeared in 2017 and introduced the *Focal Loss* function, in which the standard loss function is re-formulated and oriented to the network to put emphasis on the revision of misclassified images during training in order to improve their parameters. It has an efficiency (in COCO) of mAP@0.5=59.1% and mAP @ [0.5,0.95] = 39.1%.

B. Object Localization in a Single Image

For improvement in terms of the localization of different objects in an image, there are two groups of recently used methods:

- Refinement of *Bounding Boxes* [25]: Aims to solve the problem of having objects of unexpected scales in an image that cannot be captured by traditional methods (such as regressors). It is done through controlled iterative feedback of the detection results to the *Bounding Boxes* regressor until it converges to a suitable size and location. It should be taken into account that a very pronounced feedback of the output of an algorithm can lead to a total loss of precision.
- Loss Function Improvement for Precise Location: This addresses the two main problems of regression, which are the lack of guarantee that a small error in the regression can produce a high IoU (especially in objects with a high aspect ratio) and the second is that this IoU cannot provide location assurance [36]. To solve these drawbacks, it was proposed to include additional parameters in the Cost Function such as the IoU [33]. A *IoU Guided NMS* was also proposed to improve training and the detection stage [14]. A *Probabilistic Inference Framework* has also been developed, with the aim of making predictions related to the probability distribution of the location of the *Bounding Box* [10].

C. Detection with Rotation and Scale Changes

The rotation and the change of size or scale may be caused by elements related to the environment or by third parties, thus rotating the object or move the location in which it was as well as the zoom in or out (which may be the effect caused by the same source of the detector inputs). To solve this, different approaches have been proposed in the case of each problem.

1) Rotation: Apart from traditional solutions like *Data Augmentation* or *Multi-orientation Training*, the following three methods were developed:

- Rotation Invariant Cost Function: This solution was implemented in the 90's [29], but currently an additional variable has been added to this function that allows the characteristics of the objects to be invariant to its rotation.
- Rotation Calibration: This proposal involves making geometric changes in the objects and is generally used in multi-stage detectors, since the correlations of previous stages benefit the subsequent ones. The *STN (Spatial Transformer Networks)* are the main networks in which this proposal is applied [13].
- Rotation of ROI Pooling: Commonly, Pooling layers separate maps into multiple grids to make a more suitable size representation. This proposal manages to provide a solution to the problem of rotation by making these grids with polar coordinates [3].

2) Scale Changes: The proposed solutions for this problem are different for training and for the detection stages.

- Training: One solution involves creating *SNIPs (Scale Normalization for Image Pyramids)*, which generate detection pyramids and only back-propagate the loss on some scales [30]. Another proposal is *SNIPER (Scale Normalization for Image Pyramids Efficient Resampling)* which cuts and resizes the images in sub-regions in order to benefit the training [31].
- Detection: For this stage, the *Adaptive Zoom* is proposed, which scales the smallest objects to support the detector [9]. An improvement has also been proposed in which the scaling distribution of the different objects is predicted to re-scale them according to their distribution in an adaptive way [23].

D. Object Detection with support of ontologies

Currently, there are several works where ontologies are used to support the detection task. In the work presented in [1], the detection of relationships (which is a subsequent stage to classification) is improved through the use of ontologies in a subsequent stage to Location; Its objective is to reinforce the classification and, at the same time, obtain semantic information for use in the additional stage of Relationships. In [15], a new way of creating a framework is proposed so that a robot can move effectively. To carry out this task, use was made of numerous phases in which the detection of objects through a CNN is included but supported with the semantic information of the object from a *On-Demand Database* represented by an ontology. These classes are represented in the ontology –which allows the robot to have access to the relationships that the objects have between them– improving its recognition by using both characteristics and relationships and additional elements to classify the objects and optimize the tasks and the behavior of the robot. In [2] a framework is developed based on a *Fuzzy OWL*, which makes use of a *FuzzyDL* reasoner to be able to represent scenes in general through its fuzzy objects and its relationships; as well as being able to determine the similarity that exists between two scenes using their fuzzy descriptions. These results are achieved by assuming that the objects in the input are classified in a predetermined fuzzy class and that it has a kind of spatial relationship between the objects present.

The object detection task for *indoor* environments described in [7], is performed through a pre-trained CNN using images and videos of office objects. Subsequently, the regions of interest are obtained through a selective search and then classified as candidates through a network. The output obtained is a video showing the annotated frames. Researchers at [8] propose a new way to obtain automated data and process it,

TABLE I: Comparative chart of related works where ontologies are applied

Reference	Detection Technique	Standard	Ontology	Stage
Peursum et al., 2005 [20]	Markov Hidden Models	OWL	Activity	Post-detection
Prandí & Brumana, 2010 [22]	Fuzzy Inference	OWL	Linguistic Labels	During detection
Buoncompagni et al., 2017 [2]	-	Fuzzy OWL	Common Home Items	Used as classifier
Ding et al., 2017 [7]	Pre-trained CNN	OWL	Office items	Post-detection
Zand et al., 2016 [34]	Conditional Random Fields	OWL	Multiple classes	Post-detection
Suhkan et al., 2018 [17]	FER-CNN	OWL	Daily Life Objects	Post-detection
Ferguson et al., 2019 [8]	Extended Objects RCNN	OWL	Worksite Objects	During and post-detection
Baier et al., 2017 [1]	RCNN	OWL	Multiple classes	Post-detection
Joo et al., 2020 [15]	CNN	OWL	Indoor, outdoor, objects, people	During and post-detection
ODROM	**Pre-trained CNN (YOLOv5) + Feature Descriptors**	**OWL**	**Museums**	**Post-detection**

called *characterization of objects*. This new method consists of detecting objects in an image and obtaining semantic information from them. In their proposal, a 2D-3D object detection network was designed and applied to worksite objects and a small robotic equipment was used to detect worksite objects placed in an environment.

In [17], it is proposed a new CNN network called *FER-CNN* and it is added to the Bayesian Adaptive Recognition *framework*. The advantage of *FER-CNN* is its ability to reconstruct the hierarchy of the characteristics extracted on recognition. These reconstructed features are part of a 3D object that, later, will be connected with an ontology to explore the relationships and properties of said object. In [20], the chosen detection technique was Hidden Markov Models, as it focused on human interactions with objects rather than on the objects themselves in order to detect. They called this type of detection *indirect detection*. To carry out this technique, they used a Bayesian network to classify patches of regions in which there were object labels, which were obtained from their previous work with human interactions.

Authors in [22], propose a technique to be able to semi-automate some of the tasks required in clustering of geographic objects for recognition processes. To accomplish this task, they applied fuzzy logic as a detection tool. For a first stage, they only extracted and integrated structural information into fuzzy reasoning in order to have a more general treatment. In [34], a combination of object detection and segmentation models is presented. Authors use a Dirichlet model to

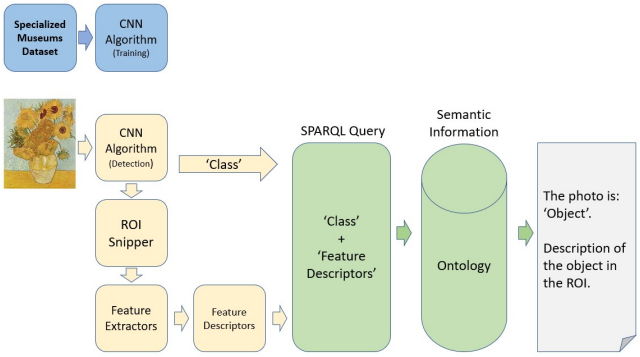


Fig. 1: Pipeline of our proposal

transform the low-level viewing space to an intermediate semantic space, in order to reduce the dimensionality of the features. Later, these characteristics are learned making use of multiple CRFs (*Conditional Random Field*) and, in turn, the segmentation of objects is used as classification when passing its inference through ontologies.

Table I summarizes the mentioned works which include ontologies in their proposals. The most widely used object detection technique is the CNN, while for ontology standards is OWL. In addition, the most common stage to apply ontologies is the post-detection. As a difference from the related work, ODROM makes use of *feature descriptors* to retrieve additional information from the ontology through a query. Those *feature descriptors* represent valuable information of the objects that ODROM uses to make a disambiguation and obtain a more accurate result.

IV. ODROM: Our Proposal

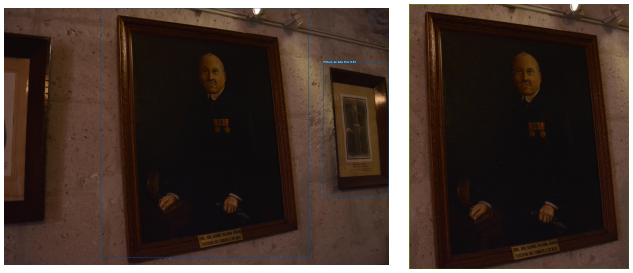
Our proposal is composed by the integration of an object detection model based on *CNN*, an intrinsic disambiguation technique of objects (based on additional characteristics or descriptors of the detected object), and an ontology that provides additional information regarding the detected objects. Fig. 1 shows the pipeline of the proposal.

A. Specialized Datased for the Application

In order to have optimal results in the development of the proposal, it is necessary to train the CNN network with a *dataset* made from images of objects related to the application in question. The goal of doing this is for CNN to learn the characteristics of the objects and be trained in the application's own classes.

The *dataset* is made up of images and labels, as well as other necessary elements that vary depending on the CNN to be used. In the case of labels, the vast majority of networks meet the following standards for labels:

- Plain text format, containing only the information with universal characters, without aesthetic or functional arrangements.
- The label document has the information related to only one object per line, considering the structure: *Class*



(a) Image of a museum classified by a CNN (b) Snipped ROI

Fig. 2: Original images of a museum

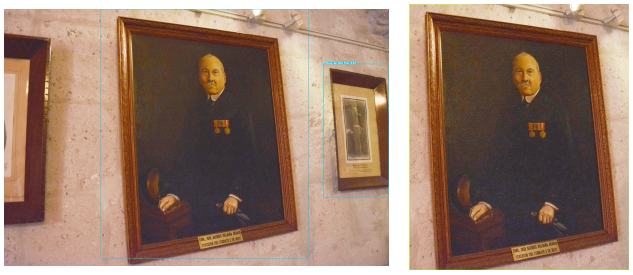


Fig. 3: Images modified by a -50% contrast and 150% brightness

number (ordinal), *Location of the center on the X axis*, *Location of the center on the Y axis*, *Width* and *Height*, in that order and with the decimal places that the labeling application is capable of granting.
- Both the image and its label have exactly the same name, with only the difference extension (e.g. .jpg and .txt).

B. CNN Training

Once the *dataset* has been assembled, the CNN training must be carried out. The results of the detection will vary depending on the performance of the computational equipment, the size of the images, the amount of them in the *dataset*, the *batch size*, the epochs, and other variables (depending on the CNN). From the training, efficiency values are obtained such as precision, recall, mAP per class, and in general, the confusion matrix, and the F1 Score can be calculated. Once the training is completed, a weight file will be obtained and can be used to detect objects in new and different images, relevant to the application.

C. ROI Cutter

To reduce the influence of the background in subsequent stages, it is necessary to remove it effectively using some technique that isolate the detected object or *ROI (Region of Interest)*. In this way, it is ensured that the subsequent stages obtain the information directly from the detected object and not from the background that surrounds it. In Fig. 2a an image classified by a CNN detector is observed, while in Fig. 2b the snip of the ROI.

D. Feature Descriptors

Each detected object has its own characteristics, intrinsic to itself that allow a description or a way to differentiate it from others within the same class. The objective of this stage is to obtain *feature descriptors* that can be used in subsequent stages to be able to establish differences and determine when a specific object was detected and when another was detected. There are a variety of *feature descriptors*, some of the most convenient being the shape, color, or semantic information of the object.

E. Ontologies in the Object Detection Task

Ontologies have great potential to organize and present information. Using them as an additional tool would enrich detection and make it more specific. In order to apply them, the class and the *feature descriptors* are used within the *query*. In case only the class is used, the ontology would return all the objects that it has instantiated within the same class. The *feature descriptors* play the role of disambiguators or filters that allow a more precise search and information on the specific object detected is obtained.

V. Experimental Validation

RUTAS project seeks to implement a mobile robot that can function satisfactorily in touristic environments. For this, it is necessary to develop computer vision, based on CNN algorithms. It is therefore necessary to create a specialized *dataset* that allows the development of an efficient model for detecting objects in museums, particularly for this work.

The dataset specialized in museums [5], was generated from photographs of different works of art, from museums in the city of Arequipa, Peru: Recoleta Museum and Municipal Museum of Arequipa.

This *dataset* is under development and will consist of 108 classes (types of artworks) and more than 3600 images. However, for the current validation, a reduced version of the dataset is used. This version consists of three classes: *Fine Art Painting*, *Ceramic as Decorative Art*, and *Fine Art Sculpture*; with 360 images, evenly distributed among classes, as a training set. Also, 111 different images are used as a validation set, also with equal amounts between classes, and multiple images with more than one class present.

The labeling of this dataset was done manually, using the *labelImg* tool, which is designed to label following the YOLO and VOC formats.

A. Pre-Trained Object Detection Algorthm Selection

To meet the project's goal of having a mobile robot capable of detecting works of art as close to real time as possible, as well as optimizing the general operation of the proposal, several models were evaluated considering detection times (expressed in *frames per second - FPS*), the efficiency and the ease of obtaining and implementation oriented to the proposal. Table II shows the algorithms evaluated based on these parameters.

[5]https://github.com/Anico18/ODROM

TABLE II: Comparison of the most recent Object Detection Algorithms based on Deep Learning

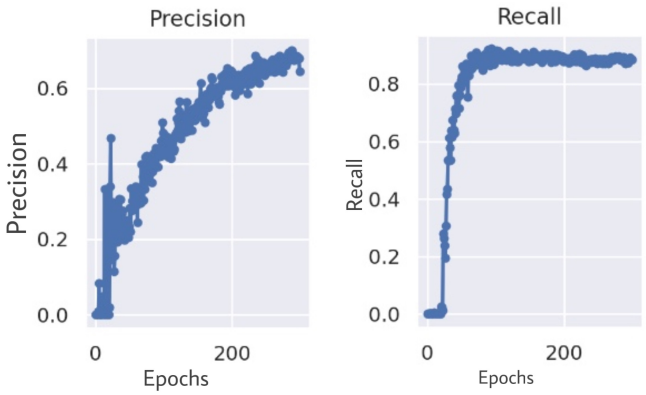
	Detectors	Efficiency	FPS
Two-stage Detectors.	RCNN (2014)	VOC07 mAP=58.5 %	0.071
	Fast RCNN (2015)	VOC07 mAP =70.0%	1.429
	Faster CNN (2015)	VOC07 mAP = 73.2 % COCO mAP@.5=42.7 % COCO mAP@[.5,.95]=21.9%	5
One-stage Detectors.	YOLO (2015)	VOC07 mAP=2.7 %(155 fps). VOC07 mAP=63.4 %(45 fps).	40 ∼155
	RetinaNet (2017)	COCO mAP@.5=59.1 % COCO mAP@[.5, .95]=39.1 %	-

From the comparison, it was concluded that YOLO is the most suitable network, due to its high speed and high enough efficiency. Added to this, the improvements that version 5 (2020)[6] has brought include better detection in spaces with many objects and higher efficiency. YOLO, as a single-stage object detection network, has high speed as one of its most outstanding characteristics. In addition to this, it must be taken into account that it does a "grid scan", in which it subdivides the image into smaller frames where it will seek to detect a single object per frame. Once one is found, it will compare its characteristics with those of the objects detected in adjacent squares and, based on the degree of similarity, it will consider it to be part of the same object or another. As a result of this, YOLO has the ability to detect multiple objects in the same image. It is necessary to mention that the speed of YOLO will decrease depending on the number of frames with which the image gets subdivided. The following describes the training and validation process of the object detection algorithm used.

1) YOLOv5 Training: The development of the YOLO algorithm and its training was carried out on a Dell Inspiron 7559 (2016) laptop, with an Intel(R) Core(TM) i7 6400HQ processor, 8GB of RAM and an NVIDIA GTX960M graphics card, and Linux 18.04 LTS as an operating system. Python was used, and the framework *PyTorch*, which includes *torchvision*, since they facilitate the implementation of pre-trained object detection algorithms.

In order to YOLO being used in the museum application, training with museum objects is necessary. Although it is a pre-trained model, it is important that it be able to extract and learn the characteristics of the objects found in these environments, in order to be able to orient it appropriately to the proposed application. The training was carried out with the dataset specialized for museums, with 300 epochs; it took about 9 hours with 40 minutes running on the graphics card. The results obtained are observed in Table III, which represents the training confusion matrix. This matrix shows the normalized value of the predictions versus the actual class during training. The matrix shows that the background of the images (interpreted as a false positive) has 67% predictions in the Ceramics class, 65% in the case of Paintings, and 92% in the case of Sculptures. However, during the tests

[6]https://github.com/ultralytics/yolov5



(a) Validation training average precision (65%)

(b) Validation training average recall (88%)

Fig. 4: Validation training

with the validation set, YOLO has no problem with false positives (as will be seen later). Fig. 4a shows the average precision obtained during training; reaching 65%. The average recall achieved in training is 88%, as shown in Fig. 4b. The mAP@0.5 and mAP@[0.5,0.95] achieved in this training process is 88% and 61% as shown in Fig. 5 and Fig. 6, respectively. The precision versus recall by class and this training's average can be seen in Fig. 7; the Ceramics as Decorative Art class have a result of 97.8%, being the best result among the three classes; while the lowest was 72.2% and belongs to the Fine Art Sculpture class.

TABLE III: Confusion Matrix obtained in the validation training

Predicted	Ceramic as decorative Art	0.32	-	0.04	0.30
	Fine Art Painting	0.02	0.35	-	0.17
	Fine Art Sculpture	-	-	0.04	0.53
	Background FP	0.67	0.65	0.92	-
		Ceramic as Decorative Art	Fine Art Painting	Fine Art Sculpture	Back-ground FN
		True			

2) Experiments with YOLOv5: To validate the training results, the test set was used. This set is made up of 36 images, in which there are objects labeled with multi-classes and others with objects that are outside the trained classes, to test if the algorithm makes an error.

During the tests, two parameters can be varied, the *IoU Threshold* and the *Confidence Threshold*; the first allows modifying the quality of the *Bounding Boxes* and the second represents the threshold that the similarity that the predicted class has to pass with the available classes so that an object can be classified within one class or another (if the threshold is exceeded in two classes, it is classified in the one with the greatest similarity). The tests were performed with the

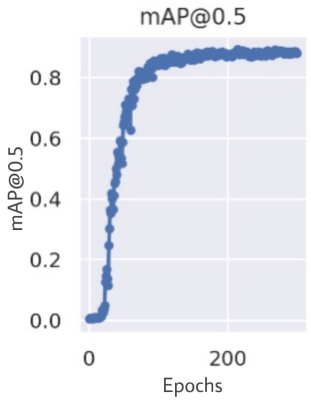


Fig. 5: Validation training average mAP@0.5 (88%)

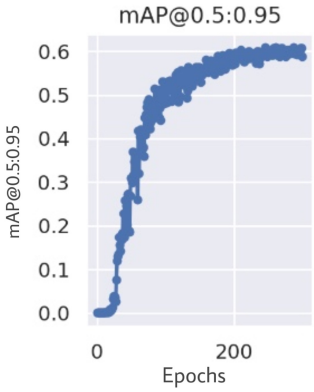


Fig. 6: Validation training average mAP@[0.5,0.95] (61%)

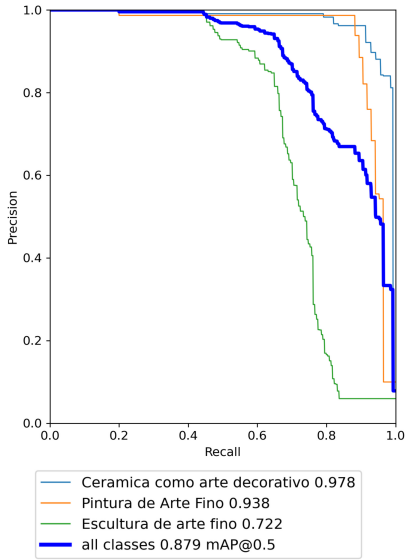


Fig. 7: Validation training Precision versus Recall Chart in general and per class

same images, *IoU Threshold*=0.8 and different *Confidence Threshold* (i.e., 0.8, 0.7, 0.6, and 0.5), in order to obtain the precision, recall, and the F1 Score, and based on them choose

TABLE IV: Detector Results using Confidence Threshold=0.8, 0.7, 0.6 y 0.5

Class	Precision	Recall	F1	Threshold
Painting	1.00	0.63	0.77	0.80
Ceramic	1.00	0.88	0.94	0.80
Sculpture	1.00	0.74	0.85	0.80
Average	1.00	0.75	0.85	0.80
Painting	1.00	0.71	0.83	0.70
Ceramic	1.00	0.93	0.97	0.70
Sculpture	1.00	0.94	0.97	0.70
Average	1.00	0.86	0.92	0.70
Painting	1.00	0.75	0.85	0.60
Ceramic	1.00	0.93	0.97	0.60
Sculpture	1.00	0.94	0.97	0.60
Average	1.00	0.87	0.93	0.60
Painting	0.98	0.76	0.86	0.50
Ceramic	1.00	0.98	0.99	0.50
Sculpture	1.00	0.94	0.97	0.50
Average	1.00	0.89	0.94	0.50

which is the most suitable threshold to make the algorithm work. The results include the comparison that each of the classes has in relation to its Precision, Recall, and F1 Score parameters and can be seen in Table IV. These results show a precision of 1 in all classes with the *Confidence Threshold* of 0.8, 0.7, and 0.6. The occurrence of an error and decreased precision happen when the *Confidence Threshold* is lowered to 0.5. The increase in recall and F1 Score is noticeable as the *Confidence Threshold* decreases, until it reaches the value of 0.5, being very low and causing a loss of precision in the class Painting.

Analyzing these results, it is observed that with *Confidence Threshold* = 0.5, the values of recall and F1 Score continue to increase, but precision in the Painting class is already beginning to decrease. This, on a small scale of images, can be insignificant, but in the application that RUTAS project develops it can become larger and critical. Maintaining a *Confidence Threshold* = 0.6 is the most appropriate, as it maximizes both the precision and the recall and F1 Score.

B. Characteristic color as Feature Descriptor

During the classification, all objects whose characteristics similarity exceed the threshold of some class, will be classified within it. This, for the task of an object detector, is considered a success if the object was correctly classified.

However, in the museum application, the single classification is not enough, since there is a great variety of objects within the same class (e.g., Painting or Sculpture). In this context, using the predominant color as the *feature descriptor* is ideal to complete this task.

Each painting has a specific color palette. In many cases, at the time, paintings can have similar colors, but never the same. In addition to this, it is known that each color has a density within the painting and that the same color has been used in many parts of the painting. Consequently, the color most present in the work (which is called *predominant or characteristic color*) can be used as a *feature descriptor* that has a very close relationship with each of the works of art.

To obtain the characteristic color of each work, a color extractor is used that defines a histogram, and through it, the color with the greatest presence in RGB code is extracted. To make it more precise and to directly obtain the color of the work of interest, the color is extracted from the objects obtained from the *ROI Cutter*. Taking into account Figs. 2a and Fig. 2b, the color of the later can be extracted as *feature descriptor*. Fig. 8 shows an example of the colors obtained from a list of snipped objects.

```
220,232,228
221,227,231
224,221,231
224,221,229
223,220,228
221,218,227
220,215,225
216,214,222
211,214,220
221,226,228
218,224,222
```

Fig. 8: Output of the Characteristic Color Extractor from a list ROIs

Even though the current validation of the proposal only implements the extraction of the characteristic color, the system can be extended, by incorporating algorithms that extract other characteristics of the detected object.

Noted that the characteristic color, as a feature descriptor, is completely reliable to any change in the location of the art in the museum since the color comes directly from the artwork and mostly ignore the surroundings due to the ROI snipper. However, this feature descriptor is not reliable in terms of lighting variance (including both natural and artificial). To overcome this difficulty, the ontology can store different RGB codes of the same artwork under different light conditions.

C. Ontology Application

The query to the ontology is carried out in order to obtain additional information about the object classified by the CNN and enrich the classification. For this, the query in SPARQL is done using the class and the characteristic color in RGB. The ontology being used is CURIOCITY Ontology [21], a semantic repository that models the information on cultural heritage and, in particular, is currently instantiated with the information from the works of art from the two museums in Arequipa considered in this work. In Fig. 9 an example diagram of the ontology query is shown.

In Fig. 10 the output of the ontology search is shown with the result of classification and extraction of characteristic color applied to the image in Fig. 2b; the result of the ontology query agrees with the actual work of art. From there, the robot will be able to extract more information related to the work of art from the ontology: author, description, year of creation, etc.

VI. General Discussion and Perspectives

Being the characteristic color an element intrinsically related to each work or artistic expression, it is a very effective

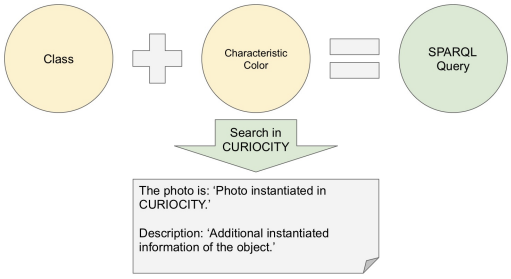


Fig. 9: CURIOCITY Query example diagram.

```
The size of the query is: 1
The photo is: http://curiocity.org/ROI_9.png
```

Fig. 10: Ontology output after query

disambiguating element. Making use of the combination of a class granted by YOLOv5, the snipping tool, and the characteristic color extractor to form a query to the ontology, it allows to greatly reduce the amount of available options of works of art that belong to the same class, reaching even to identify the actual work. However, its application may be limited by possible light modifications within the room when capturing the image and with monochrome works.

It is possible that, within larger searches and with different works, there are cases in which the characteristic color is not enough to do the total disambiguation in the query. To fix this, a second-level disambiguation technique must be applied. It is obvious that with more additional characteristics that can be extracted from the detected object, the better the filter that is applied to the query to the ontology will be and the more precise the final result will be.

This first experience with ODROM demonstrates the feasibility of combining detection algorithms, feature extractors, and ontologies, to improve the result of the object detector classification, i.e., identify a specific object within the same class (e.g., *Portrait of the Crnl. José Alcides Villalba Araujo - Vencedor del Combate 2 de Mayo* within the "Painting" class) and offer more information about the object detected in the image (e.g., description, author, year of creation, current owner), which is located stored in the semantic repository.

By using the Specialized Dataset made, and specially to the combination with the CNN, the feature descriptor, and the ontology, a much better and accurate result can be obtained. Thanks to this, receiving a precise description of the art classified is possible and a better experience in museums (in the application) can be achieved.

VII. Conclusions

This paper describes ODROM, a pipeline for detecting objects in images, based on ontologies and applied to museums. ODROM is based on a CNN algorithm to detect objects in images (YOLOv5 on validation tests), an extractor of the characteristic color of the detected object and an ontology of works of art. We show that the enrichment of object detection through the information that can be provided by the

ontology is highly feasible and applied in contexts in which a classification is not enough to identify a specific object in an image. Furthermore, ontologies allow to store much more information about objects of interest that can be retrieved from the results obtained from the detector and extractor.

We are currently working on extending the feature extractor from objects with second-level disambiguation techniques (extracting features other than color) and considerations of light effects in images.

Additionally, this same work will be developed with a much larger specialized dataset, both in number of images and classes. Likewise, everything presented will be integrated to the robotic simulator *ROS (Robot Operating System)* and tested on a robot within a real controlled environment.

ACKNOWLEDGEMENT

This research was supported by the FONDO NACIONAL DE DESARROLLO CIENTÍFICO, TECNOLÓGICO Y DE INNOVACIÓN TECNOLÓGICA - FONDECYT as executing entity of CONCYTEC–PERU under grant agreement no. 01-2019-FONDECYT-BM-INC.INV in the project RUTAS: Robots for Urban Tourism Centers, Autonomous and Semantic-based (Robots para centros Urbanos Turísticos Autónomos y basados en Semántica).

REFERENCES

[1] Stephan Baier, Yunpu Ma, and Volker Tresp. Improving visual relationship detection using semantic modeling of scene descriptions. In *International Semantic Web Conference*. Springer, 2017.

[2] Luca Buoncompagni, Fulvio Mastrogiovanni, and Alessandro Saffiotti. Scene learning, recognition and similarity detection in a fuzzy ontology via human examples. 2017.

[3] B. Cai, Z. Jiang, H. Zhang, Y. Yao, and S. Nie. Online exemplar-based fully convolutional network for aircraft detection in remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, 2018.

[4] Francesco Capezio, Fulvio Mastrogiovanni, Antonello Scalmato, Antonio Sgorbissa, Paolo Vernazza, Tullio Vernazza, and Renato Zaccaria. Mobile robots in hospital environments: an installation case study. In *ECMR*, 2011.

[5] Chenyi Chen, Ming-Yu Liu, Oncel Tuzel, and Jianxiong Xiao. R-cnn for small object detection. In *Asian conference on computer vision*. Springer, 2016.

[6] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[7] Xintao Ding, Yonglong Luo, Qingying Yu, Qingde Li, Yongqiang Cheng, Robert Munnoch, Dongfei Xue, and Guorong Cai. Indoor object recognition using pre-trained convolutional neural network. In *International Conference on Automation and Computing*. IEEE, 2017.

[8] Max Ferguson, Seongwoon Jeong, Kincho Law, and M Asce. Worksite object characterization for automatically updating building information models. 2019.

[9] Mingfei Gao, Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Dynamic zoom-in network for fast object detection in large images. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[10] Spyros Gidaris and Nikos Komodakis. Locnet: Improving localization accuracy for object detection. In *conference on computer vision and pattern recognition*, 2016.

[11] Ross Girshick. Fast r-cnn. In *international conference on computer vision*, 2015.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *transactions on pattern analysis and machine intelligence*, 2015.

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*. 2015.

[14] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *European Conference on Computer Vision*, 2018.

[15] Sung-Hyeon Joo, Sumaira Manzoor, Yuri Goncalves Rocha, Sang-Hyeon Bae, Kwang-Hee Lee, Tae-Yong Kuc, and Minsung Kim. Autonomous navigation framework for intelligent robots based on a semantic environment modeling. *Applied Sciences*, 2020.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012.

[17] Sukhan Lee, Ahmed M Naguib, and Naeem Ul Islam. 3d deep object recognition and semantic understanding for visually-guided robotic service. In *International Conference on Intelligent Robots and Systems*, 2018.

[18] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *international conference on computer vision*, 2017.

[20] Patrick Peursum, Geoff West, and Svetha Venkatesh. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In *International Conference on Computer Vision*. IEEE, 2005.

[21] Alexander Pinto-De la Gala, Yudith Cardinale, Irvin Dongo, and Regina Ticona-Herrera. Towards an Ontology for Urban Tourism. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, page 1887–1890, New York, NY, USA, 2021.

[22] Federico Prandi and Raffaella Brumana. Semi-automatic objects recognition process based on fuzzy logic. In *International Conference on Personal Satellite Services*. Springer, 2010.

[23] Siyuan Qiao, Wei Shen, Weichao Qiu, Chenxi Liu, and Alan Yuille. Scalenet: Guiding object proposal generation in supermarkets and beyond. In *IEEE International Conference on Computer Vision*, 2017.

[24] Yu Qing-xiao, Yuan Can, Fu Zhuang, and Zhao Yan-zheng. Research of the localization of restaurant service robot. *International Journal of Advanced Robotic Systems*, 2010.

[25] Rakesh N Rajaram, Eshed Ohn-Bar, and Mohan M Trivedi. Refinenet: Iterative refinement for accurate object localization. In *International Conference on Intelligent Transportation Systems*, 2016.

[26] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *conference on computer vision and pattern recognition*, 2016.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 2015.

[28] Kanade Rowley, Baluja. Human face detection in visual scenes. In *In Advances in Neural Information Processing Systems*, 1996.

[29] Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. Transformation invariance in pattern recognition—tangent distance and tangent propagation. In *Neural networks: tricks of the trade*. Springer, 1998.

[30] Bharat Singh and Larry S. Davis. An analysis of scale invariance in object detection snip. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[31] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*. 2018.

[32] Mohammad Abu Yousuf, Yoshinori Kobayashi, Yoshinori Kuno, Keiichi Yamazaki, and Akiko Yamazaki. Social interaction with visitors: mobile guide robots capable of offering a museum tour. *Transactions on Electrical and Electronic Engineering*, 2019.

[33] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *ACM international conference on Multimedia*, 2016.

[34] Mohsen Zand, Shyamala Doraisamy, Alfian Abdul Halin, and Mas Rina Mustaffa. Ontology-based semantic image segmentation using mixture models and multiple crfs. *Transactions on Image Processing*, 2016.

[35] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. Is faster r-cnn doing well for pedestrian detection? In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision*. Springer International Publishing, 2016.

[36] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. 2019.