Stacked Generalization for scene analysis and object recognition

Lorenzo Peppoloni∗, Massimo Satler†, Emanuel Luchetti‡, Carlo Alberto Avizzano∗ and Paolo Tripicchio†

∗ PERCeptual RObotics Laboratory
† Centro Gustavo Stefanini
TeCIP Institute, Scuola Superiore Sant’Anna.
Email: l.peppoloni@sssup.it

Abstract—The problem of object recognition and detection has been largely addressed by the robotics community, since its importance both in mapping and manipulation problems. One possible approach for the recognition task is to assume a specific a-priori knowledge of the objects possibly present in a scene. In this framework, this paper presents a novel technique for object detection and recognition based on Stacked Generalization (SG) method developed by Wolpert in 1992. The innovation of the proposed technique is the introduction of SG classification method to perform a multi-layer object recognition fusing heterogeneous spatial and color data acquired with an RGB-D camera. To improve the accuracy and the robustness of the system to environmental variability, we introduce a second layer classifier. Its goal is to evaluate and weights the results of the first layer classifiers, thus combining and improving the overall classification performance. This technique has a low computational cost and is suitable for on-line applications, such as robotic manipulation or automated logistic systems. To validate the presented approach experimental tests have been carried out and results are reported.

I. INTRODUCTION

Robots, as humans, need to be endowed with the capability of exploring and understanding the real environment they are interacting with. In order to increase the complexity of the tasks that mobile or humanoid robots can perform in real environments, high semantic understanding of the world [1], [2] is required. In particular, object detection and recognition appear to be a fundamental parts of the process. Objects and obstacles localization is crucial both for understanding the environment and adapting robot behaviors to it [3]. Moreover, in object manipulation tasks, the force to be applied is generally derived according to the properties of the grasped object, thus heavily relying on recognizing the manipulated object. One of the main focus of object recognition research can be formalized as follow: given a set of known objects \( O = \{O_1, O_2, \ldots, O_N\} \), recognize those objects in a complex visual scene. This kind of problem is referred by the research community as instance recognition and can be addressed as a classification problem. In the literature it has been widely studied, exploiting several classification methods among which the more significant ones are: template matching, statistical classification and the ones based on machine learning techniques [4]. Template Matching techniques [5] are the simplest and the earliest developed and they are based on the concept of similarity measure. Each time a recognition must be performed, the feature vector, extracted from the acquired data, must be compared with all the templates of each object stored in the database (the template are stored as feature vectors as well). The acquired data are classified as belonging to the object with the most similar template. Even if this method is effective in some specific application domains, it is computationally demanding because the number of comparison operations to be performed in real-time can grow fast with both (a) the number of the objects and (b) the number the templates (object scans) stored for each object. Moreover, with this method, generalization comes at price of dramatically increasing the number of templates recorded in the database.

In statistical classification methods [6] data acquired from object scans are represented with features as well, but the object model is built with a probability density function for each class. The probability density function measures the probability of an element (object scan) of a class to have a given feature vector. Classification is here achieved using Bayes theorem to find the probability that the feature vector of the acquired data belong to a class (object).

Statistical classification methods are known to give low accuracy in classification especially when the feature space has a large dimensionality. In machine learning classification approaches [7], the object model is automatically learned from a given dataset by training procedures. The goal of the training phase is to find a decision boundary in the feature vector space by minimizing certain error function. One of the most widely used algorithm for classification is the Support Vector Machine (SVM) classifier [8], [9]. The main advantages of SVM classifiers are the generalization capabilities (not related to the feature space cardinality) and the small number
of training data required for the learning phase. In object recognition tasks, SVM classifiers are proved to have low computational cost in the prediction phase, meaning that they are suitable for on-line applications.

In this paper we exploit the accuracy of the SVM introducing a novel multilevel architecture with a stacked generalization approach [10]. SG is a machine learning technique used to minimize the generalization error of one or more generalizers. The idea in our approach consists in the introduction of a second space of generalization to deduce the biases of the original generalizers. The second layer input is the output of the first layer of generalizers, and its output is the generalization guess. This approach has been successfully exploited in computer vision in [11], where the authors present a two layer approach for obstacle detection. In [12] the authors follow the same approach to classify five types of feature for semantic learning of rooms. In this paper we propose a similar approach to classify known objects in a scene, acquired with a RGB-D sensor. The raw sensor data are fed to a three stages pipeline, consisting on preprocessing, object detection and classification steps. The output is the ID of every detected object.

To test the accuracy and the robustness of the presented approach, experimental tests have been carried on in different environmental conditions.

The paper is organized as follow. Section I provides an overview of the architecture of the developed processing pipeline; Section III, IV and V present the details of the pipeline steps. Finally, Section VI presents the experimental results. In Section VII our concluding remarks are reported.

II. METHODOLOGY

The approach presented in this paper relies on the pipeline architecture shown in Figure 2. This pipeline is used to analyze data acquired scanning a scene with a RGB-D sensor. In the scene there are \( M \) objects taken from a set \( O = \{ O_1, O_2, \ldots, O_N \} \) of \( N \) known objects. The goal is to detect the \( M \) objects therein, and to recognize them, i.e. to label each one of them as one of the \( O_i \) objects. It is to be noted that the same pipeline is used both during the training phase of the system and for the actual object recognition step. The main processing steps are the data preprocessing, the segmentation of the point cloud in \( M \) clusters \( \{ C_1, C_2, \ldots, C_M \} \) representing the objects detected in the scene, the extraction of the feature vector \( \Phi_j \) for each cluster, and finally the classification of each cluster \( C_j \). Classifying the cluster means assigning each detected object to a class corresponding to one of the objects in \( O \), through its feature vector, thus achieving the object recognition \( C_j = O_i \).

In particular, the three main processing blocks are:
- Data Preprocessing
- Object Detection
- Object Classification

Preprocessing steps deals with raw data processing, such as downsampling and noise filtering. The output is a 6D (color and 3D position) point cloud. In the Object Detection step the point cloud is segmented to extract every single object from the whole acquired scene. Moreover in this step features extraction is performed for each detected object. The output is a descriptor in the features space for each detected object. Classification step is based on SG to assign each detected object to the proper class, thus performing a recognition of the object. A detailed description of each step will follow.

III. PREPROCESSING

The first processing step on raw data is a range filtering aimed at removing RGB-D point outliers, due to sensor errors or points outside the interest range (distance from the sensor computed along the \( Z \) coordinate in sensor reference frame). Filtered point cloud is then downsampled, by Voxel Grid Filtering Algorithm: all points inside a 3x3x3 mm volume (called Voxel) are replaced by their centroid (averaging color and position coordinates). Downsampling filtering is mainly required to reduce computational costs and sensor noise effect.

IV. OBJECT DETECTION

The Object Detection phase (Figure 3) is composed of two fundamental steps. The first is the segmentation, in which the point cloud is processed to obtain the clusters of points belonging to every sensed object. The second step is the feature extraction, in which a feature vector for each cluster is obtained.

![Fig. 2. Processing pipeline, used for both training and prediction phase. Preprocessing step filters and downsamples data, Object Detection steps deals with scene point cloud segmentation and features extraction. Lastly Object Classification step assigns to each cluster the corresponding object label.](image)

![Fig. 3. Object Detection step. First the preprocessed point cloud is segmented, and then the features are extracted from the clusters.](image)
A. Segmentation

The main goal of this step is removing all the points not belonging to any object and clustering the remaining points into $M$ clusters. Each cluster corresponds to a different perceived object. Since all the objects are assumed to be laying on a plane, the ground-plane is detected using RANSAC-based model fitting algorithm [13]. The algorithm searches for the 3D primitive plane which best fits the point cloud and identifies the set $S_p$ of the points belonging to it. After plane detection, the point cloud is divided in two sets: $S_1$, corresponding to the prism above the plane whose base is $S_p$, and $S_2$ as its complement on the whole point cloud. All the objects are supposed to be in $S_1$, thus at this point $S_2$ is discarded with a simple passage filter on the proper axes coordinates. A Statistical Filter algorithm [14] is applied to remove residual points belonging to the plane $S_p$ and due to sensor noise. Finally, the object segmentation algorithm subdivides the remaining points of $S_1$ into subclusters $\{C_1, C_2, C_3, \ldots\}$ associated to each single object. All Point Cloud elaborations are supported by Point Cloud Library (PCL) [15]. Figure 4 shows the resulting segmentation from the raw point cloud.

B. Features Extraction

Since we focus on recognizing specific objects, color and geometrical properties depending on object shape and viewpoint have been chosen as features. Regarding colors, the RGB coordinates are transformed into Hue Saturation Value (HSV) color space which, as demonstrated in [16], [17], is invariant with respect to light intensity. The description of the three features used follows.

1) Hue Histogram: For each RGB-D point belonging to the same object $C_j$, the color components are extracted and converted to the HSV space. Hue channel values of the points in $C_j$ are then histogrammed with 128 bins. The resulting histogram feature will provide the classifier with all the color components of each view of an object.

2) Object Dominant Hues: Each detected object $C_j$ can be characterized observing its dominant colors, corresponding to local maxima in the histogram curve. The dominant colors information are already included in Hue Histogram, but we chose to extract those features to provide the network with information encoded in a different way: the histogram indicates how much each hue is frequent, while the dominant indicates which color is more present, thus being a qualitative information, rather than quantitative. To extract Object Dominant Hues feature, the Hue histogram is differentiated according to the trivial formula $D_i = H_i - H_{i-1}$, then variations from positive to negative values are searched to detect the dominant components of the histogram. The indexes related to the three highest peaks (local maxima) of the Hue-color histogram are chosen as Dominant Hue feature. The Object Dominant Hues feature vector is completed with Hue mean and standard deviation values to provide the classifier with a better characterization of the detected object. Experiments have proven that this choice provides better classification performances.

3) Viewpoint Features Histogram: Viewpoint Features Histogram (VFH) was developed by Rusu et al. in [18], it is a descriptor for 3D point cloud data that encodes geometry and viewpoint information on a 308 components vector. The first part of the histogram encodes the angle between the normal to the surface in each point and the view direction of the centroid, while the second part encodes the angles of each normal with respect to a specific coordinate frame depending from the viewpoint and the position of the point relative to the centroid.

V. Classification

The main goal of the classification step is to perform the recognition of the objects detected in the previous step. The problem of recognition is reduced to classification, where every detected cluster in the scene is associated to the corresponding object label. In particular each class corresponds to an object to be recognized, and the instance of a class corresponds to a cluster acquired from the scene containing the object.

The classification is composed of two different layers, as shown in Fig. 5.

The features are passed as input to a multi-level combination of SVM classifiers. A first set of SVM classifiers (Level-0 experts) is trained to classify the detected object using only a subset of the feature vector. Level-0 outputs are then used as inputs for an additional SVM classifier specifically trained to learn the biases of all the experts and to give the most probable response. Every layer is selectively weighted in order to adapt the classification process to the different possible scene conditions. This approach is also called Late Fusion
scheme to differentiate it from classical Early Fusion scheme where all the features extracted are aggregated in a global vector used as input for one only SVM classifier.

The training phase for this particular structure, requires two consecutive steps. First Level-0 classifiers need to be trained. Once the Level-0 is ready, the different subset of the training set is used as input for the Level-0 to obtain the answers that will form the training set for the classifier of the Level-1. In this way, we let the system being able to generalize on Level-0 output without overfitting the training set. All SVMs are trained performing a 5-fold cross validation algorithm and a grid search is performed to estimate the best parameters for each of them.

VI. EXPERIMENTAL RESULTS

The system has been tested on a dataset composed of five different daily objects, shown in Figure 6. RGB-D data has been acquired using a Microsoft Kinect camera. Due to the constraints of the low-cost RGB-D sensor used in the experiment, we made the following assumptions without loss of generality:

- Objects are assumed to be relative Lambertian, they cannot be transparent or highly reflective
- Object detection and recognition task is assumed for indoor environment only and without direct sunlight exposition
- Operating distance is assumed between 0.5 m and 2.0 m far from the sensor device

To create the training set we have collected 360 samples for each of the five objects, using the setup shown in Figure 7. Each object was sequentially rotated by 36 steps of 10 degrees and for each step we acquired ten frames in different light conditions (sun light, artificial light, cloudy day, morning, afternoon, evening, etc).

After the training phase, the system performance has been assessed, presenting the objects in different poses and light conditions. Results are reported in Table I, where the results of every Level-0 classifier are compared to the final system accuracy. Figure 8 shows the performances of the system in different conditions. In particular the system is capable of recognizing the objects in several light conditions, and when partial occlusions are taken into account. In every image the system GUI is reported, where top-left view shows real time RGB image, top-right view shows real time raw point cloud, bottom-left view shows elaboration results and bottom-right view shows classification results and the coordinates of the objects centroids.

![Image](image.png)

**Fig. 6.** The objects used to validate the system.

![Image](image.png)

**Fig. 7.** Experimental setup for the point cloud acquisition, for the data set. Each object was rotated by 36 steps of 10 degrees and for each step we acquired ten frames in different light conditions.

![Image](image.png)

**Fig. 5.** SG approach for the classification step. The features computed in the detection step are used as input to the two layers of SVM classifiers for classification. The output are the objects ID

---

**Table I**

<table>
<thead>
<tr>
<th>Object</th>
<th>Hue Dominant SVM</th>
<th>Hue Hist SVM</th>
<th>VHF SVM</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lego</td>
<td>94.9 ± 2.1%</td>
<td>94.9 ± 2.4%</td>
<td>94.9 ± 2.2%</td>
<td>94.9 ± 2.2%</td>
</tr>
<tr>
<td>Wall-E</td>
<td>90.1 ± 1.4%</td>
<td>92.9 ± 2.9%</td>
<td>92.5 ± 2.0%</td>
<td>93.7 ± 2.1%</td>
</tr>
<tr>
<td>Orzo</td>
<td>89.5 ± 3.0%</td>
<td>91.1 ± 2.1%</td>
<td>92.8 ± 2.1%</td>
<td>93.1 ± 2.2%</td>
</tr>
<tr>
<td>Teapot</td>
<td>93.0 ± 1.4%</td>
<td>95.5 ± 2.8%</td>
<td>95.6 ± 1.1%</td>
<td>96.1 ± 1.6%</td>
</tr>
<tr>
<td>Access Point</td>
<td>94.7 ± 1.7%</td>
<td>95.6 ± 0.6%</td>
<td>94.7 ± 1.6%</td>
<td>96.2 ± 0.9%</td>
</tr>
</tbody>
</table>

The results confirm how SG technique improves the classification performance, when compared to the single classifiers composing the Level-0 stage. As far as computation time is concerned the most expensive phase of the pipeline (Figure 2) is the Object Detection, where a great amount of data has to be processed to obtain the objects clusters $C_j$ to be classified.
Fig. 8. System performances in different scenarios. The images show the system GUI. In each image top-left view shows real time RGB image, top-right view shows real time raw point cloud, bottom-left view shows elaboration results and bottom-right view shows classification results and objects centroid coordinates.

Classification itself is the cheapest phase taking less than the 3% of the total process time.

The system has been also tested in extreme conditions, see Figure 9, where objects have been detected and classified in night environment (complete dark), obtaining correct classification for each of the five objects acquired by the sensor. This particular case highlights the strength of SG method: only the Level-0 classifier working on geometric features (VFH) receives an informative input. In such an extreme case, Level-1 generalizes on Level-0 output, discarding the two incorrect answers from Color-based features classifiers and using only the correct answer given by SVM classifier based on VFH feature. The results of the test are reported in Table II.

TABLE II

<table>
<thead>
<tr>
<th>Object</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lego</td>
<td>69%</td>
</tr>
<tr>
<td>Wall-E</td>
<td>98%</td>
</tr>
<tr>
<td>Orzo</td>
<td>93%</td>
</tr>
<tr>
<td>Teapot</td>
<td>89%</td>
</tr>
<tr>
<td>Access Point</td>
<td>92%</td>
</tr>
</tbody>
</table>

Stacked Generalization method using RGB-D data. We have demonstrated the goodness of our approach with experimental results obtained using a low cost commercial RGB-D sensor to learn and classify a test set composed of everyday objects in different environmental condition. Future efforts will focus on improving the point clouds segmentation phase to correctly identify objects in very challenging configurations, i.e. large occlusions and objects stacking (without spatial separation). To obtain a better classification accuracy we will study the implementation of new kind of visual and geometric features extraction methods to add more experts to the Level-0 stage of our Classification step. Finally, a method to estimate the 6D pose of objects (position and orientation) will be studied to obtain a more complete environment mapping system.

VII. CONCLUSIONS AND FUTURE WORK

We have proposed a new approach to the Object Detection and Recognition problem based on the application of

REFERENCES