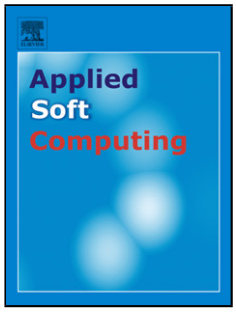


Title: Semi-supervised 3D Object Recognition through CNN Labeling

Author: José Carlos Rangel Jesus Martínez-Gómez Cristina Romero-González Ismael García-Varea Miguel Cazorla

PII: S1568-4946(18)30055-3
DOI: https://doi.org/doi:10.1016/j.asoc.2018.02.005
Reference: ASOC 4696

To appear in: *Applied Soft Computing*

Received date: 24-5-2017
Revised date: 15-1-2018
Accepted date: 2-2-2018

Please cite this article as: José Carlos Rangel, Jesus Martínez-Gómez, Cristina Romero-González, Ismael García-Varea, Miguel Cazorla, Semi-supervised 3D Object Recognition through CNN Labeling, *<![CDATA[Applied Soft Computing Journal]]>* (2018), https://doi.org/10.1016/j.asoc.2018.02.005

Semi-supervised 3D Object Recognition through CNN Labeling

José Carlos Rangel[a,c,*], Jesus Martínez-Gómez[b], Cristina Romero-González[b], Ismael García-Varea[b], Miguel Cazorla[a]

[a] *Institute for Computer Research, University of Alicante., P.O. Box 99. 03080, Alicante, Spain.*
[b] *Computer Systems Department, University of Castilla-La Mancha, Spain.*
[c] *RobotSIS, Universidad Tecnológica de Panamá, Panamá.*

Abstract

Despite the outstanding results of Convolutional Neural Networks (CNNs) in object recognition and classification, there are still some open problems to address when applying these solutions to real-world problems. Specifically, CNNs struggle to generalize under challenging scenarios, like recognizing the variability and heterogeneity of the instances of elements belonging to the same category. Some of these difficulties are directly related to the input information, 2D-based methods still show a lack of robustness against strong lighting variations, for example. In this paper, we propose to merge techniques using both 2D and 3D information to overcome these problems. Specifically, we take advantage of the spatial information in the 3D data to segment objects in the image and build an object classifier, and the classification capabilities of CNNs to semi-supervisedly label each object image for training. As the experimental results demonstrate, our model can successfully generalize for categories with high intra-class variability and outperform the accuracy of a well-known CNN model.

Keywords: Object Recognition, Deep Learning, Object Labeling, Machine Learning

*Corresponding author
Email address: jcrangel@dccia.ua.es (José Carlos Rangel)

1. Introduction

Object recognition is a challenging research area of growing interest in recent years due to its applicability in fields such as autonomous robotics and scene understanding. The research in this field has been stimulated by the appearance of more sophisticated cameras, as well as the capability of learning from a vast amount of images.

Nowadays, we can clearly differentiate between object recognition approaches based on 3D or 2D image information. Conventional 3D object recognition approaches deal with problems such as occlusions, holes, noise, and rotation, translation or scale invariance [1, 2, 3, 4, 5, 6]. The use of 3D information is computationally expensive, and demands enormous storage facilities. This last point, in conjunction with the cost of manually labeling 3D images, makes it difficult to release interesting large object recognition datasets with 3D labeled objects. The lack of datasets with sufficient 3D information of a vast range of objects [5, 6] represents a persisting problem, as some of the most outstanding machine learning techniques, especially Deep Learning (DL), require huge labeled sequences for their effective training. However, 2D object recognition has benefited from the release of DL techniques, and more specifically Convolutional Neural Networks (CNNs), to obtain promising results in recent years [7, 8, 9, 10, 11]. Actually, CNNs trained from huge datasets like ImageNet present high generalization capabilities using a varied kind of objects and heterogeneous images.

Despite the latest advances in 2D object recognition, there are still several drawbacks to the use of perspective images, such as the difficulty of working in dark environments. This point may be a requirement for some robotic applications, as surveillance or inspection. Moreover, the geometry of the object may be relevant to avoid false positives. For example, an object with an unusual color might not be correctly classified, but this issue could be successfully solved using a classification method based on the 3D shape of the object.

The work presented in this article takes advantage of previous research in both 2D and 3D object recognition to propose semi-supervised classification solutions. To this end, we exploit the labeling capabilities of current DL-based 2D approaches [12, 13, 14], but relying on 3D input images to deal with the already described drawbacks.

Figure 1 presents our approach graphically. Initially, objects are detected in 3D images (encoded as point clouds) by means of a clustering algorithm.

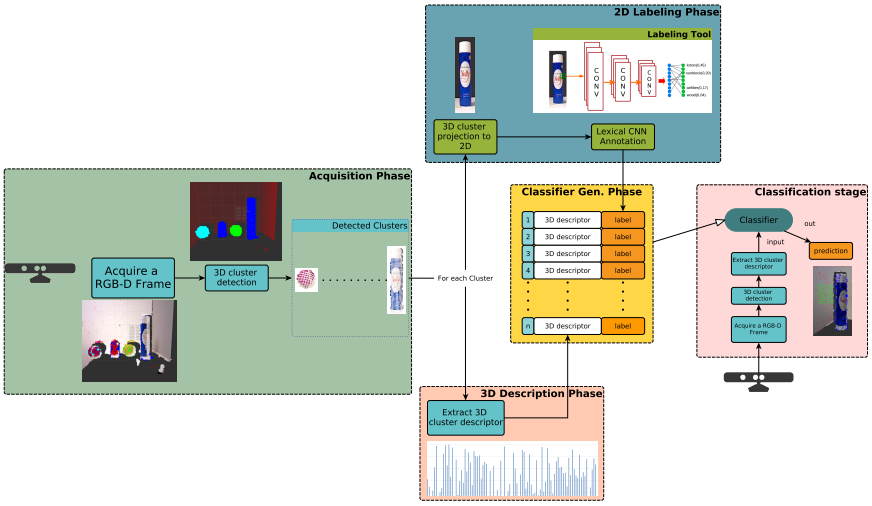


Figure 1: Graphical overview of the proposal. Green and blue boxes indicate processing of 2D (RGB images) and 3D (RGB-D point clouds) information respectively.

Then, each cluster is projected into its perspective image, and it is labeled using an CNN-based lexical annotation tool.

At the same time, the 3D data from each cluster are processed to extract a 3D descriptor. The set of descriptor-label pairs define the training dataset that will serve to build a classification model, which will carry out the effective object recognition process.

Our proposal successfully exploits the knowledge included in previously trained CNN models to annotate 3D input images, that feed a semi-supervised 3D object classifier. But, as a consequence of relying on an external labeling tool to generate the training set, some samples may be wrongly labeled. However, we have performed an extensive evaluation to assess the robustness of the proposal. According to the experimental results, we found a remarkable performance in the best possible scenario (most of the training samples are correctly labeled) and a competitive performance in other randomly selected scenarios (the training samples are selected randomly without considering their ground truth label).

The rest of the paper is organized as follows. Section 2 details state-of-the-art related to object recognition. Next, in Section 3, the details of our proposal are presented. In Section 4, we describe how the experimental set was selected, and we present the experimental results obtained. Section 5

presents a discussion of the experimental results. Finally, in Section 6 the main conclusions of this work are outlined.

2. Related Work

There exist several approaches to perform 3D object recognition, most of them based on feature matching. One example is presented in the review by Guo et al. [5], where the authors detail the basic stages for the recognition, and also describe the available datasets, feature descriptors, and keypoint detectors that have been widely used in a vast majority of the studies. Another example is Tombari et al. [1] where the authors study the recognition ability of stereo algorithms. Similarly, the solution presented in Asari et al. [2] evaluates several 3D shape descriptors to determine their feasibility in 3D object recognition tasks.

Using a combination of images and depth maps for cluttered scenes, as proposed by Hinterstoisser et al. [15], the authors obtain positive recognition results. In order to detect free-form shapes in a 3D space, a Hough Voting algorithm was proposed by Tombari and Di Stefano [16] yielding good recognition rates. Pang and Neumann [17] propose to combine 3D local features with machine learning techniques in order to detect objects in different kinds of images. Using a novel Global Hypothesis Verification algorithm, Aldoma et al. [3] refine the obtained results, discarding false positive instances. Based on the former work, Rangel et al. [4] carried out a study aimed at testing whether a Growing Neural Gas (GNG) could reduce the noise in scene point clouds and therefore manage to recognize the objects present in the 3D scene.

In recent years, DL has been widely applied to solve the 3D object recognition problem. For example, ModelNet[1] aims at finding DL-based approaches to produce accurate classification models. In this challenge, participants were provided with a dataset of 10 or 40 categories of 3D CAD objects. Different solutions are presented for this challenge, like the proposal presented by Wu et al. [18] that represents 3D shapes as probability distributions using binary variables on a grid of 3D voxels.

Nowadays, a trend in recognition is to mix several representations and CNN designs in order to generate sufficiently discriminative information about the objects. For instance, the DeepPano approach presented in [19]

[1] http://modelnet.cs.princeton.edu/

uses a cylindric projection of the object to build a panoramic view that feeds a CNN. In [20], the use a 3D Generative Adversarial Network and a 3D CNN for classification tasks is proposed. Other proposals, like VoxNet [21] and PointNet [22], combine CNNs with occupancy grid representations.

The use of neural networks for classification problems has inspired the modification or mixing of existing defined structures in order to achieve better results. For instance, the FusionNet [23] approach uses two Multiview Convolutional Neural Network (MVCNN) fed with a pixel representation and a volumetric representation. This MVCNN is used also by Su et al. [24], but with a renderized image collection. The Voxception ResNet (VRN) proposed by Brock et al. [25] merges a ResNet architecture with inceptions blocks. Finally, the LonchaNet technique, proposed by Gomez-Donoso et al. [26] uses an ensemble of GoogLeNet architectures that are feeding with slices of the original object.

The use of CNNs for object classification has introduced several advantages in the field that may be summarized as follows. Firstly, the existence of several DL frameworks and the ease of use of these allows the users to straightforwardly develop their own models based on their specific requirements. Among these frameworks, we can mention TensorFlow [27] [2], Theano[3], MxNet[4] and Caffe [14][5]. Secondly, owing to these frameworks, the user community has created a massive library of models that others can freely use. These models were also trained with different image datasets, such as ImageNet [12], Places [28] or a combination of both. With this in mind, we find another leverage point, the generalization capability, as a consequence of the use of the CNN to learn features from a vast number of images. Moreover, the use of these pre-trained models avoids having an enormous amount of data (dataset of images) in order to train a classifier. Consequently, applications demand less storage space and processing time.

CNNs have been used in object detection tasks and challenges such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [12, 13]. This challenge focuses on processing 2D images in order to achieve Scene Classification and object detection in the ImageNet dataset. In the latest editions, winning teams have based their solutions on the use of CNNs.

[2]https://www.tensorflow.org/
[3]http://deeplearning.net/software/theano/
[4]http://mxnet.io/
[5]http://caffe.berkeleyvision.org/

Among the works that employ the CNNs for object detection and recognition, we can mention Girshick et al. [9], where the R-CNN or Region with CNN features is presented; then Ren et al. [29] improved this proposal by employing a network for region proposal and another for classification; Bui et al. [7] modify an AlexNet architecture by replacing the fully connected layer with a RNN for detecting objects in images; and Qu et al. [30] where the authors used RGB and depth images with saliency detectors to feed a CNN and produce a saliency map for object detection.

CNNs have been also used as supervised object detectors, like in [10, 11, 9, 8], but these detectors require the bounding box annotation for training the models. Meanwhile, the proposal by Tang et al. [31] focuses on developing object detectors by transference of visual and semantic knowledge.

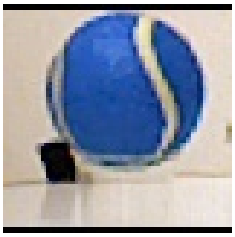
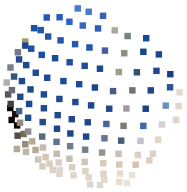
Others approaches, such as [32, 33, 34, 35], involving the use of CNN for object detection, are known as weakly supervised methods. These methods train a CNN detector using images that have been labeled at image level and without the bounding box information of the objects. Therefore, these approaches employ the CNN as a feature extractor.

Although the previous 2D object recognition approaches provide good classification results, there are still some problems to be solved. For example, objects with high intra-class variability can be problematic when the classification is solely based on the 2D information. In Table 1 we illustrate this problem. The same object with a rare color would be incorrectly classified with a state-of-the-art 2D classifier. However, the 3D shape is the same in both images, so this information could be use to correctly classify the object.

3. Semi-supervised 3D Object Recognition through CNN Labeling

The aim of our proposal is to build an object recognition system based on 3D descriptors, but generated in a semi-supervised way. To this end, our technique generates a training set trusting in the labels provided by an external agent, namely a CNN classifier previously trained with large quantities of perspective images. In addition to the labels, we use 3D fixed-dimensionality features extracted from input point clouds, aiming to combine procedures from 2D to 3D scopes. The training set would include a new instance for every cluster detected in the input point cloud, as they can feasibly correspond with an object.

Table 1: Predictions predicted by a CNN classifier for two tennis ball with different color, also the point cloud of every object is shown in order to see the similitudes of their morphology.

Image	Point Cloud	Predictions	Probability
		pick	0.498
		digital clock	0.117
		tennis ball	0.078
		stopwatch	0.039
		digital watch	0.038
		knee pad	0.477
		pick	0.038
		spatula	0.030
		rock beauty	0.029
		barrel	0.026

Our proposal involves a training stage from the previously generated training set. We should take into account that the labeling system may initially present classification errors. Therefore, we assume our classification model must deal with errors derived from the labeling tool. Figure 1 shows an overall flowchart of the proposal. The solution presented in this article can be divided in four different phases: Acquisition, 2D Labeling, 3D Description, and Classifier Generation.

3.1. Acquisition Phase

This phase begins with the acquisition of an RGB-D image in point cloud format Then, this point cloud is clustered using an Euclidean clustering algorithm in order to detect possible objects clusters. Figure 2 (left) shows an example of input RGB image with its corresponding point cloud, and the detected clusters.

3.2. 2D Labeling Phase

Once clusters have been identified, every one is projected into a 2D RGB image. Then, the next step is to assign a representative lexical label to the object image (see Figure 2 (right)).

To do that, the cropped image is passed to a labeling system. This process provides a set of labels with an associated probability value and for every cropped image we select the label with the maximum probability value.

This process could be carried out by a human expert or by any system or tool able to provide a lexical label from a set of predefined labels, that is, following a supervised or semi-supervised approach, respectively. In our proposal, the lexical labels will be assigned by a pre-trained CNN model with the AlexNet [36] architecture in a semi-supervised fashion, as this is a state-of-the-art classifier for RGB images.

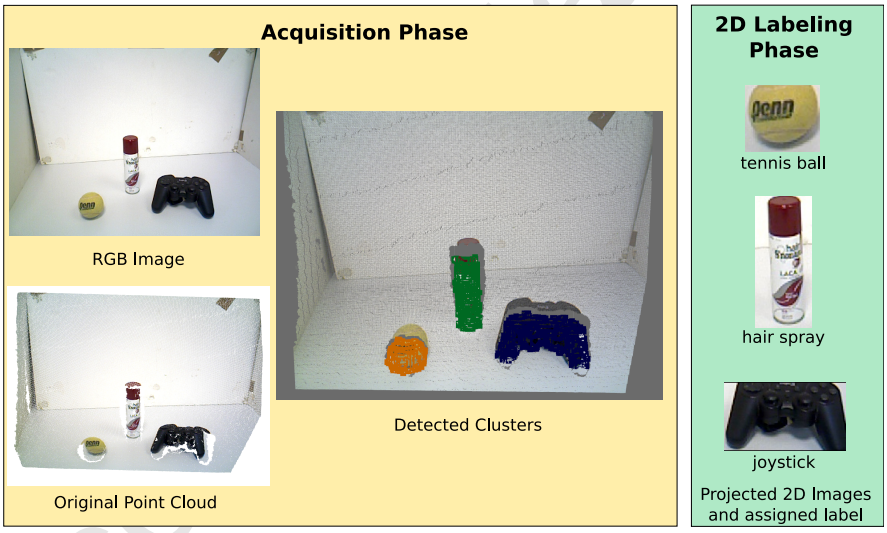


Figure 2: Example of the Acquisition (left) and 2D Labeling (right) Phases of the proposal.

3.3. 3D Description Phase

At the same time, the description phase takes the identified clusters in the point cloud and computes a 3D global descriptor for each. The descriptor selected for this phase is the Ensemble of Shape Functions (ESF)[37]. This descriptor is a 3D point cloud global descriptor which consists of a combination of three different shape functions that result in 10 concatenated 64-bin

histograms. It is based on three shape functions describing distance (the distance between two randomly selected points), angle (the angle enclosed by two lines created from three randomly selected points), and area distributions (the area of the triangle formed by three randomly selected points). It also classifies each of the values into three classes based on where the connecting lines between points reside: *on* the object surface, *off* the surface and *mixed* (partly *on* and *off*). So the sub-histograms used to form the ESF are: 3 *(on, off, mixed)* for angle, 3 (*on, off, mixed*) for area, 3 *(on, off, mixed)* for distance, and a final one for the ratio of line distances between *off* and *on* parts of each line considered.

In contrast to other 3D descriptors, ESF does not require normal information, which makes it robust to noise and partial occlusions. Each histogram contains 64 bins, and the final dimension of the descriptor is 640. Figure 3 shows the three histograms (*on, off and mixed*) obtained for the angle, area, and distance function, as well as the additional histogram, and the structure of the generated ESF descriptor.

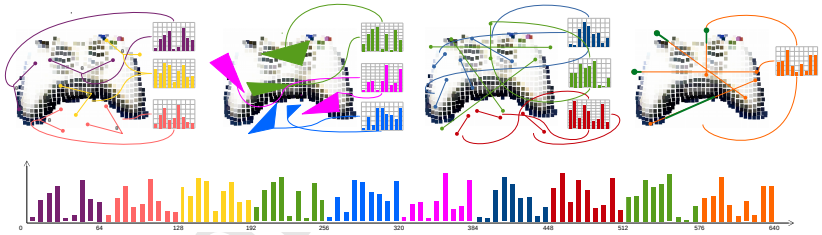


Figure 3: Generation of the ESF Descriptor.

3.4. Classifier Generation Phase

Once the above phases have finished, the next step is to train a classifier using the samples of the object instances. Each sample consists on a pair of the 3D descriptor and its assigned lexical label.

We propose the use of a Support Vector Machine (SVM) with a Radial Basic Function (RBF) kernel as a classification model. Given the training data, a hyperplane is defined to optimally discriminate between different categories. If a linear classifier is used, SVM constructs a line that performs an optimal discrimination. For the non-linear classifier, kernel functions are used, which maximize the margin between categories.

4. Experimental Results

4.1. Experimental Dataset

Our proposal relies on the labels obtained from a pre-trained CNN model to preform the semi-supervised classification. Usually, these models are trained over the ILSVRC 2012 dataset, which contains hundreds of 2D images for each of its 1,000 categories. However, this dataset does not contain 3D data, so we have captured RGB-D images of different objects present in the ILSVRC 2012 dataset from different viewpoints.

Specifically, we have captured 500 RGB-D images for each object instance, with 3 instances per object, and 8 objects (*carton*, *computer keyboard*, *hair spray*, *joystick*, *loafer*, *ocarina*, *remote control* and *tennis ball*). In Figure 4 we show a sample of each object instance.

The Algorithm 1 resumes the training set acquisition procedure for the approach. Here, we can observe the continuous application of the Acquisition, 2D Labeling and 3D Description phases over an input cloud with objects.

Algorithm 1: Data Adquisition Procedure

input : $PointCloud \leftarrow$ Data captured by the RGB-D camera
output: TrainingInstances $\leftarrow \emptyset$
Data: ClusterList $\leftarrow$ Extract Clusters in $PointCloud$
forall *cluster* C_i *in ClusterList* **do**
  $I_C \leftarrow$ 2D Projection of C_i
  $l_i \leftarrow$ Predict Label for I_C
  $D_{C_i} \leftarrow$ Extract 3D Features of C_i
  $Instance_i \leftarrow$ Make a pair with (D_{C_i}, l_i)
  Add $Instance_i$ to TrainingInstances
end forall
Return TrainingInstances

In the 2D Labeling Phase, the lexical labels available for every image are those of the 8 labels that represent the selected objects shown in Figure 4. Since the CNN model outputs a total of 1,000 labels, we used only the confidence value produced for those labels corresponding with the objects selected for the study. Therefore, the assigned label will correspond to the selected object with highest confidence value. Figure 5 shows the result for the labeling phase of each object instance. Here, the blue numbers indicate the

Figure 4: Samples for each instance of the 8 object categories used in the experiments.

amount of correctly labeled images. These results will be used to construct the training and test sets for the experiments.

4.2. Experimental Setup

In order to validate the proposal, we must define the objects that will form the training and test datasets. Concretely, we are going to select two instances of each object for training and the other instance for test. This results in a split of 1,000 samples per object (8,000 samples) for training and 500 samples per object (4,000 samples) for test.

The best-case scenario considers a training/test split where, for each object, the two instances with the highest number of correctly labeled samples are selected for training and the other instance for test. This scenario represents a the worst case for a 2D classifier, where it cannot adequately generalize

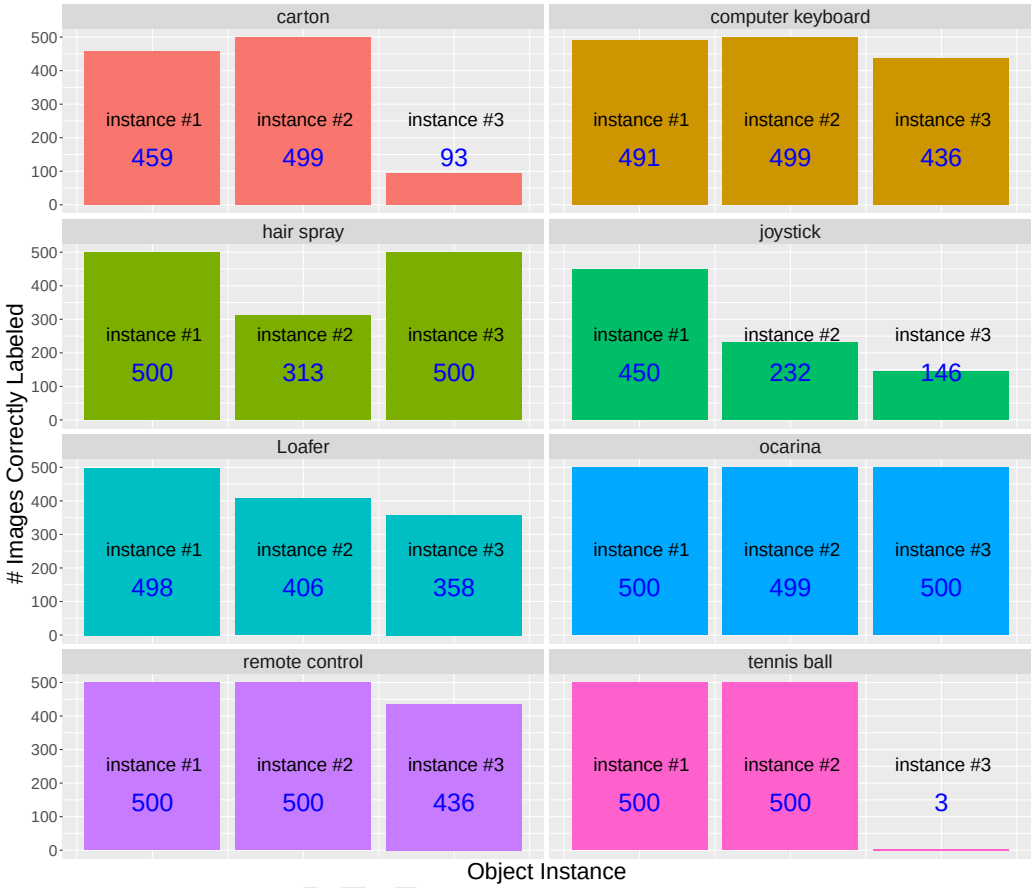


Figure 5: Number of correctly labeled frames for each object instance using the CNN labeling tool.

due to the visual differences in instances of a same object. This train/test split, then, is designed to outline the robustness that a 3D-based classifier can provide for a vision system deployed in a real scenario.

To also present a more realistic scenario for testing the robustness of the proposed 3D object classification system, we have considered 30 different scenarios where the two instances of each object are randomly selected for training, and the other instance is used for test. This might be considered as a leave-one-instance-out validation approach.

The results will be presented using well-known metrics to evaluate the performance of classification experiments [38]. Also, owing to the multi-class nature of this classification problem, the metrics were calculated by the 1-

vs-all method. Therefore, we can analyze the metrics for the classifiers by object category. Specifically, and given the number of true positives (tp), false positives (fp), true negatives (tn), and false negatives (fn), these metrics are defined as:

$$A = \frac{tp}{tp + fp + tn + fn}; \;\; P = \frac{tp}{tp + fp}; \;\; R = \frac{tp}{tp + fn}; \;\; S = \frac{tn}{tn + fp}$$

where A is the accuracy, P the precision, R the recall or sensitivity or true positive rate, and S is the specificity or true negative rate.

Finally, the SVM classification has been performed with the libsvm library [39].

In the following, we will compare the result obtained with our method, against a pre-trained AlexNet [36] CNN as this is a state-of-the-art classifier for 2D images. **We have used the proposed architecture with the same hyperparameters, so we refer the reader to the original paper for more details on its implementation. In addition, this is the model used for lexical labeling in our approach, so we can directly assess whether our semi-supervised system is able to generalize its results under challenging conditions.**

4.3. Best-case Scenario Results

In this scenario, the accuracy obtained by our proposed classification model is 93.5%, whereas the accuracy obtained for the same test dataset with the CNN model was 57.1%. Hence, our method outperforms the results produced by the CNN classifier.

The confusion matrices for this experiment are shown in Figure 6, here we can see the success of our system in identifying the majority of the objects used in the experiments while the CNN has problems with some of the them due to their intra-class variability. Consequently, this experimental results determine that the proposed pipeline is able to improve the recognition results of a state-of-the-art CNN model in object classification.

Table 2 displays evaluation metrics previously mentioned for the CNN model and our proposed classifier. Values in bold highlight the best of the two methods for each evaluation metric. Interestingly, comparing the accuracy and recall values, we can observe that our method obtains better results than the CNN classifier for all objects. Specially, it is remarkable how in the tennis ball example, the CNN model fails to identify most of the test samples (it

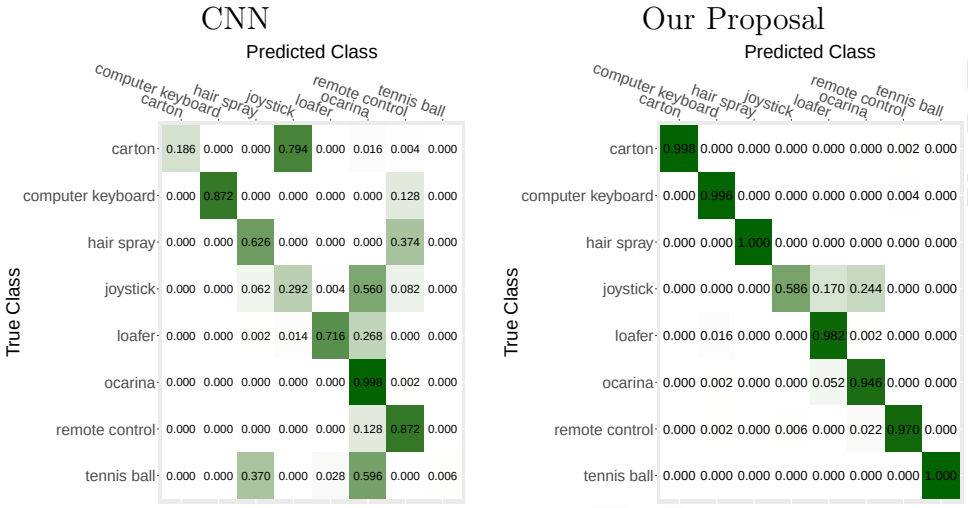


Figure 6: Confusion matrix for the 2D CNN classifier (left) and our 3D object recognition system (right).

Table 2: Evaluation metrics for the best-case scenario.

Category	**Specificity**		**Precision**		**Recall**		**Accuracy**	
	CNN	Our	CNN	Our	CNN	Our	CNN	Our
carton	1.000	1.000	1.000	1.000	0.186	**0.998**	0.593	**0.999**
computer keyboard	**1.000**	0.997	**1.000**	0.980	0.872	**0.996**	0.936	**0.997**
hair spray	0.938	**1.000**	0.591	**1.000**	0.626	**1.000**	0.782	**1.000**
joystick	0.885	**0.999**	0.266	**0.990**	0.292	**0.586**	0.588	**0.793**
loafer	**0.995**	0.968	**0.957**	0.816	0.716	**0.982**	0.856	**0.975**
ocarina	0.776	**0.962**	0.389	**0.779**	0.998	**0.946**	0.887	**0.954**
remote control	0.916	**0.999**	0.596	**0.994**	0.872	**0.970**	0.894	**0.985**
tennis ball	1.000	1.000	1.000	1.000	0.006	**1.000**	0.503	**1.000**

only obtains a 0.006 recall and 0.503 of accuracy), while our 3D-shape based method is able to identify all these test samples correctly.

Figures 7 and 8 show the Precision-Recall curve (left) and the ROC curve

(right) for our semi-supervised 3D object recognition system and the CNN model, respectively. These charts graphically display the values of the Table 2. The area under the curve of the charts shows the appropriateness of classifying objects based in their 3D features, following the approach presented in this paper.

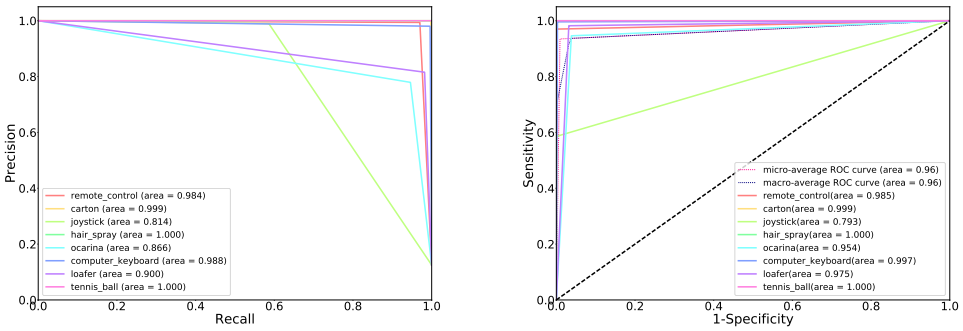


Figure 7: Precision-Recall curve (left) and ROC curve (right) for the 3D object recognition system.

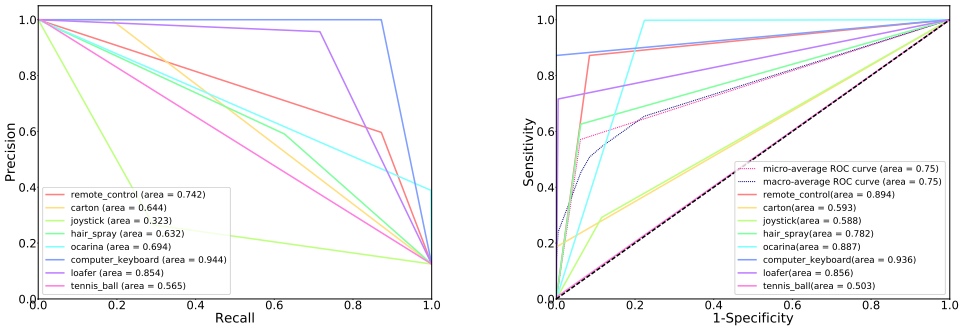


Figure 8: Precision-Recall curve (left) and ROC curve (right) for the CNN classifier.

Finally, Tables 3 and 4 present a success and a fail case, respectively, for our approach. In Table 3, due to the visual appearance of the image, the blue *tennis ball* was misclassified by the CNN model as an *ocarina*, but was correctly classified by our system based on its 3D shape. On the other hand, the fail case presented in Table 4, where a *computer keyboard* is misclassified as a *remote control* is striking.

Table 3: Example of success case for our system.

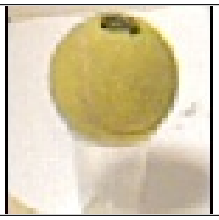
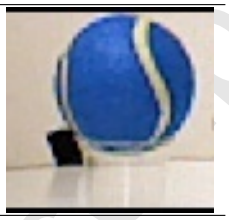
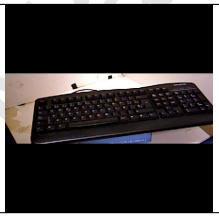
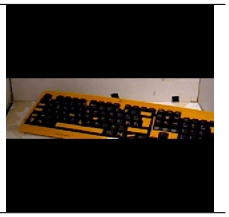
	Training Instances		Test Instance
			
Ground Truth	tennis ball	tennis ball	tennis ball
CNN Class	tennis ball	tennis ball	ocarina
Predicted Class			tennis ball

Table 4: Example of fail case for our system.

	Training Instances		Test Instance
			
Ground Truth	comp. keyboard	comp. keyboard	comp. keyboard
CNN Class	comp. keyboard	comp. keyboard	comp. keyboard
Predicted Class			remote control

4.4. Results with Random Train/Test Splits

Figure 9 presents the accuracy values obtained for 30 randomly selected scenarios. The right side shows the accuracy obtained with our 3D system for each experiment, while the left side shows the corresponding accuracy with the CNN model. The highlighted labels (green) indicate the experiments where the our classification model outperforms the CNN classifier.

Our proposal obtained a mean accuracy of 84.55% ($\sigma = 4.90$) while the state-of-the-art classifier obtained a mean accuracy of 84.40% ($\sigma = 7.70$). These results show that, in general conditions, our system will have a similar performance to a state-of-the-art classifier, even if it is trained with mislabeled samples due to the semi-supervised nature of the proposal. While, under the adequate conditions, it can clearly overcome the drawbacks of current state-of-the-art object classification systems, as illustrated in our best-case scenario in Section 4.3.

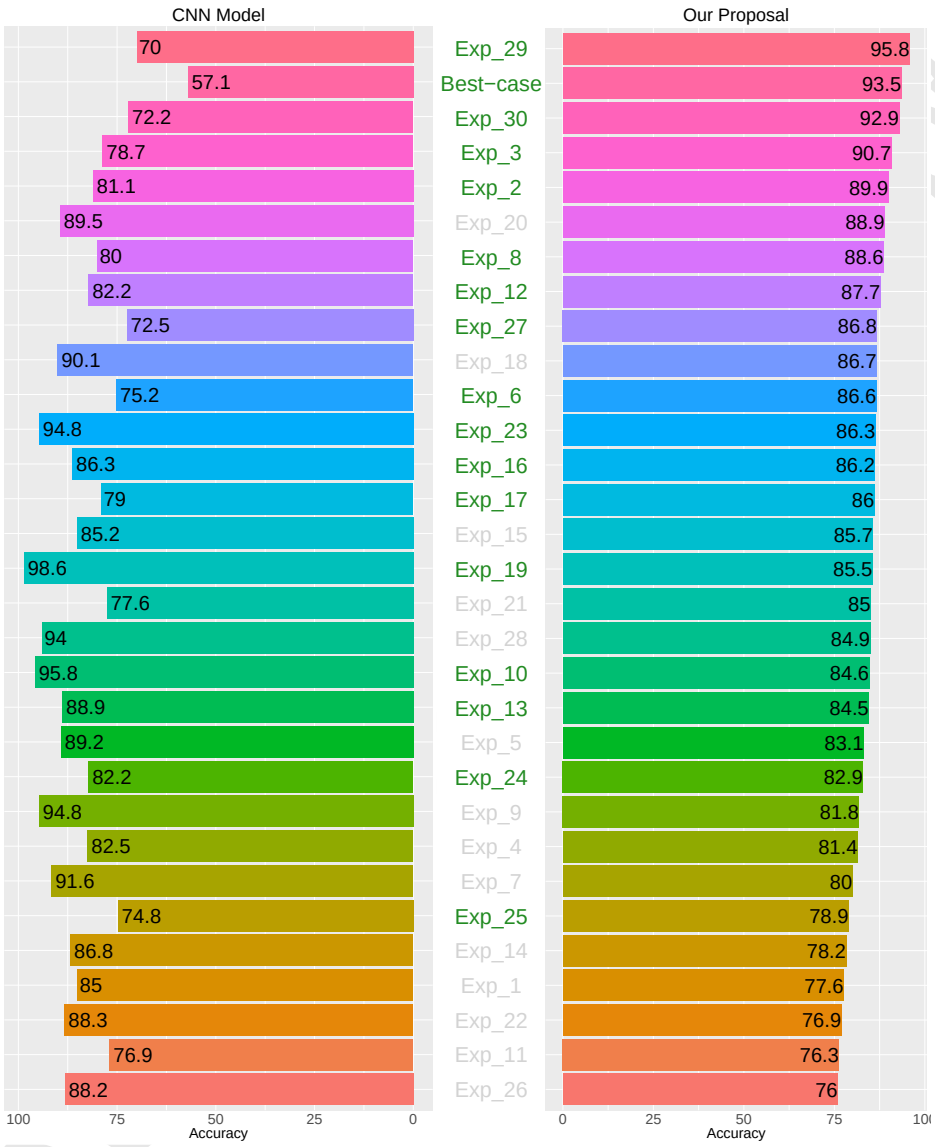


Figure 9: Accuracy obtained by every combination.

Tables 5 and 6 present a success and a fail case, respectively, for the experiment with the lowest accuracy rate. In Table 5, a *loafer* is correctly classified by the classification model, although there are instances in the train dataset which are misclassified by the labeling tool, this example illustrates the robustness of our proposal. On the other hand, Table 6 shows an scenario

where the *carton* object was not recognized by neither the labeling tool nor the 3D classification model. In this case, the misclassified train samples biased the learned model.

Table 5: Success case from the lowest accuracy experiment

	Training Instances		**Test Instance**
			
Ground Truth	loafer	loafer	loafer
Labeling Tool Class	ocarina	loafer	loafer
Predicted Class			loafer

Table 6: Fail case from the lowest accuracy experiment

	Training Instances		**Test Instance**
			
Ground Truth	carton	carton	carton
Labeling Tool Class	remote control	carton	remote control
Predicted Class			comp. keyboard

Table 7 presents the composition of the training and test dataset for the scenarios with the highest and lowest accuracy, as well as the best-case combination.

5. Discussion

The aim of the research presented in this article was to find a novel way to recognize objects based on the use of 3D information. This type of recognition has to deal with several drawbacks such as dark environments, occlusions, as well as the lack of sufficient object datasets for training. Therefore, this

Table 7: Highest and lowest accuracy scenarios description.

Lowest Accuracy Combination	**Highest Accuracy Combination**	**Best-case Combination**
Exp_26 Acc=76.0%	Exp_29 Acc=95.8%	Baseline Acc=93.5%
Training Instances		
carton 2	carton 2	carton 1
carton 3	carton 3	carton 2
computer keyboard 2	computer keyboard 1	computer keyboard 1
computer keyboard 3	computer keyboard 2	computer keyboard 2
hair spray 1	hair spray 1	hair spray 1
hair spray 2	hair spray 3	hair spray 3
joystick 1	joystick 1	joystick 1
joystick 2	joystick 3	joystick 2
Loafer 2	Loafer 1	Loafer 1
Loafer 3	Loafer 2	Loafer 2
ocarina 1	ocarina 1	ocarina 1
ocarina 2	ocarina 3	ocarina 3
remote control 1	remote control 3	remote control 1
remote control 2	remote control 2	remote control 2
tennis ball 1	tennis ball 1	tennis ball 1
tennis ball 3	tennis ball 2	tennis ball 2
Test Instances		
carton 1	carton 1	carton 3
computer keyboard 1	computer keyboard 3	computer keyboard 3
hair spray 3	hair spray 2	hair spray 2
joystick 3	joystick 2	joystick 3
Loafer 1	Loafer 3	Loafer 3
ocarina 3	ocarina 2	ocarina 2
remote control 3	remote control 1	remote control 3
tennis ball	tennis ball 3	tennis ball 3

paper proposes the use of a 2D labeling system for assigning categories to objects and describing them with a 3D feature descriptor in order to train a classification model. Our results support the initial premise and demonstrate the suitability of the solution to the problems described.

The experiments developed for the validation of the proposal show that the procedure developed has the capacity to recognize unseen objects based only on their 3D features, without the need to train a classifier using a highly similar instance of the object. Our results also demonstrate that the use of 3D data helps to overcome certain difficulties faced by 2D classification systems, such as the difference in visual appearance between objects in the training and test sets. **The benefits of our system are specially highlighted when we take into account that we improve the accuracy of a well-known**

CNN model while the exactly same model is used for labeling the training samples of the 3D classifier. So, it is not too far-fetched to assume that using a different CNN model with better base accuracy would also render better results for our 3D classifier.

The CNN model has shown its capacity to accurately classify 2D object images. Nevertheless, the use of these DL-based labeling systems generates errors in the datasets used for training the classification models. However, these errors do not significantly affect the accuracy of the proposed method when a reduced number of misclassified instances are used for training the model. Hence, the use of the 3D features enables us to build a robust object classifier **that, with appropriate training instances (correctly labeled), can greatly improve state-of-the-art solutions**. Additionally, our approach avoids the need to train a CNN model, which requires a huge annotated dataset and high computational resources.

The findings in this research facilitate the adaptation of the proposed method for using in challenging areas, for example, places with different lighting conditions. Furthermore, the generalization capability of the classifier allows a small degree of independence for the systems, robots or platforms that select the proposed pipeline for classification issues. This type of applications would be able to run in real time, as the average classification time for one sample is 14 ms.

6. Conclusions

In this work, we have proposed the use of an external labeling tool for assigning a lexical label to a cluster of points detected in a point cloud. Then, with these clusters, a classifier is trained to recognize instances of the clusters that are difficult to classify using a 2D classification method.

Experimental results show that even though some instances were misclassified by the labeling tool, our classification model is able to recognize the objects based on their 3D features. **Additionally, our experiments demonstrate that, given the adequate conditions, the trained 3D classifier easily outperforms well-known CNN models.**

Results also show the advantages of using combined 2D and 3D procedures in order to make the most of the generalization capabilities of 2D classification models as well as the morphology information provided by the 3D data. **This merging of 2D and 3D methods gives our proposal a high discriminative capacity in object recognition.**

As future work, we plan to integrate the classification model in a mobile robot, to detect objects in an environment or location, and make it possible to grasp or manipulate them after being successfully classified.

7. Acknowledgements

This work has been partially sponsored by the Spanish Ministry of Economy and Competitiveness under grant number TIN2015-65686-C5-3-R. It has been also supported by the Spanish Government TIN2016-76515-R Grant, supported with Feder funds. Cristina Romero-González is funded by the MECD grant FPU12/04387. José Carlos Rangel is funded by the IFARHU grant 8-2014-166 of the Republic of Panamá.

8. Bibliography

[1] F. Tombari, F. Gori, L. Di Stefano, Evaluation of stereo algorithms for 3d object recognition, in: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, pp. 990–997.

[2] M. Asari, U. Sheikh, E. Supriyanto, 3d shape descriptor for object recognition based on kinect-like depth image, Image and Vision Computing 32 (2014) 260 – 269.

[3] A. Aldoma, F. Tombari, L. Di Stefano, M. Vincze, A global hypotheses verification method for 3d object recognition, in: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (Eds.), Computer Vision ECCV 2012, volume 7574 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2012, pp. 511–524.

[4] J. C. Rangel, V. Morell, M. Cazorla, S. Orts-Escolano, J. García-Rodríguez, Object recognition in noisy rgb-d data using gng, Pattern Analysis and Applications (2016) 1–16.

[5] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, 3d object recognition in cluttered scenes with local surface features: A survey, Pattern Analysis and Machine Intelligence, IEEE Transactions on 36 (2014) 2270–2287.

[6] A. Garcia-Garcia, J. Garcia-Rodriguez, S. Orts-Escolano, S. Oprea, F. Gomez-Donoso, M. Cazorla, A study of the effect of noise and occlusion on the accuracy of convolutional neural networks applied to 3d object recognition, Computer Vision and Image Understanding (2017).

[7] H. M. Bui, M. Lech, E. Cheng, K. Neville, I. S. Burnett, Object recognition using deep convolutional features transformed by a recursive network structure, IEEE Access PP (2017) 1–1.

[8] R. Girshick, Fast r-cnn, in: The IEEE International Conference on Computer Vision (ICCV).

[9] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14, IEEE Computer Society, Washington, DC, USA, 2014, pp. 580–587.

[10] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated recognition, localization and detection using convolutional networks, CoRR abs/1312.6229 (2013).

[11] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems 28, Curran Associates, Inc., 2015, pp. 91–99.

[12] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, pp. 248–255.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, Imagenet large scale visual recognition challenge, International Journal of Computer Vision (IJCV) 115 (2015) 211–252.

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, arXiv preprint arXiv:1408.5093 (2014).

[15] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, V. Lepetit, Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes, in: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 858–865.

[16] F. Tombari, L. Di Stefano, Object recognition in 3d scenes with occlusions and clutter by hough voting, in: Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on, pp. 349–355.

[17] G. Pang, U. Neumann, Training-based object recognition in cluttered 3d point clouds, in: 3D Vision - 3DV 2013, 2013 International Conference on, pp. 87–94.

[18] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: A deep representation for volumetric shapes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912–1920.

[19] B. Shi, S. Bai, Z. Zhou, X. Bai, Deeppano: Deep panoramic representation for 3-d shape recognition, IEEE Signal Processing Letters 22 (2015) 2339–2343.

[20] J. Wu, C. Zhang, T. Xue, W. T. Freeman, J. B. Tenenbaum, Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling, arXiv preprint arXiv:1610.07584 (2016).

[21] D. Maturana, S. Scherer, Voxnet: A 3d convolutional neural network for real-time object recognition, in: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, IEEE, pp. 922–928.

[22] A. Garcia-Garcia, F. Gomez-Donoso, J. Garcia-Rodriguez, S. Orts-Escolano, M. Cazorla, J. Azorin-Lopez, Pointnet: A 3d convolutional neural network for real-time object class recognition, in: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 1578–1584.

[23] V. Hegde, R. Zadeh, Fusionnet: 3d object classification using multiple data representations, CoRR abs/1607.05695 (2016).

[24] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3d shape recognition, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 945–953.

[25] A. Brock, T. Lim, J. Ritchie, N. Weston, Generative and discriminative voxel modeling with convolutional neural networks, arXiv preprint arXiv:1608.04236 (2016).

[26] F. Gomez-Donoso, A. Garcia-Garcia, S. Orts-Escolano, J. Garcia-Rodriguez, M. Cazorla, Lonchanet: A sliced-based cnn architecture for real-time 3d object recognition, in: 2017 International Joint Conference on Neural Networks (IJCNN).

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[28] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, in: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 487–495.

[29] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems (NIPS).

[30] L. Qu, S. He, J. Zhang, J. Tian, Y. Tang, Q. Yang, Rgbd salient object detection via deep fusion, IEEE Transactions on Image Processing PP (2017) 1–1.

[31] Y. Tang, J. Wang, B. Gao, E. Dellandréa, R. Gaizauskas, L. Chen, Large scale semi-supervised object detection using visual and semantic knowledge transfer, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2119–2128.

[32] C. Wang, K. Huang, W. Ren, J. Zhang, S. Maybank, Large-scale weakly supervised object localization via latent category learning, IEEE Transactions on Image Processing 24 (2015) 1371–1385.

[33] H. O. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, T. Darrell, On learning to localize objects with minimal supervision, in: T. Jebara,

E. P. Xing (Eds.), ICML - 31st International Conference on Machine Learning, volume 32 of *JMLR Workshop and Conference Proceedings*, JMLR, Beijing, China, 2014, pp. 1611–1619.

[34] H. Bilen, M. Pedersoli, T. Tuytelaars, Weakly supervised object detection with convex clustering, in: CVPR.

[35] H. Bilen, M. Pedersoli, T. Tuytelaars, Weakly supervised object detection with posterior regularization, in: BMVC.

[36] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. Burges, L. Bottou, K. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1097–1105.

[37] W. Wohlkinger, M. Vincze, Ensemble of shape functions for 3d object classification, in: Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on, IEEE, pp. 2987–2992.

[38] D. M. W. Powers, Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation, Journal of Machine Learning Technologies 2 (2011) 37–63.

[39] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology (TIST) 2 (2011) 27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Highlights

- A novel approach for 3D object recognition is proposed.
- The proposal relies on deep learning pre-trained models for image annotation.
- Mixing 2D and 3D techniques for processing data improve recognition capabilities.
- Proposal can get over errors introduced by the labeling tool.
- Experimental results prove the effectiveness of the proposed procedure.

2D Labeling Phase
Labeling Tool
C O N V
C O N V
C O N V
lotion(0,45)
sunblock(0,20)
airfilter(0,17)
wood(0,04)
3D cluster projection to 2D
Lexical CNN Annotation
Acquisition Phase
Detected Clusters
Acquire a RGB-D Frame
3D cluster detection
For each Cluster
Classifier Gen. Phase
1 3D descriptor label
2 3D descriptor label
3 3D descriptor label
4 3D descriptor label
n 3D descriptor label
Classification stage
Classifier
out
input
prediction
Extract 3D cluster descriptor
3D cluster detection
Acquire a RGB-D Frame
3D Description Phase
Extract 3D cluster descriptor