In: Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), pages: 2068–2075, Rio de Janeiro, Brazil. IEEE, July, 2018.

Point Cloud Object Recognition using 3D Convolutional Neural Networks

Marcelo Borghetti Soares
Knowledge Technology
Department of Informatics
University of Hamburg
Vogt-Koelln-Str. 30, 22527 Hamburg, Germany
marcelo.borghetti@gmail.com

Stefan Wermter
Knowledge Technology
Department of Informatics
University of Hamburg
Vogt-Koelln-Str. 30, 22527 Hamburg, Germany
wermter@informatik.uni-hamburg.de

***Abstract*—With the advent of RGB-D technology, there was remarkable progress in robotic tasks such as object recognition. Many approaches were developed to handle depth information, but they work mainly on 2.5D representations of the data. Moreover, the 3D-data handling approaches using Convolutional Neural Networks developed so far showed a gap between volumetric CNN and multi-view CNN. Therefore, the use of point clouds for object recognition has not been fully explored. In this work, we propose a Convolutional Neural Network model that extracts 3D features directly from RGB-D data, mixing volumetric and multi-view representations. The neural architecture is kept as simple as possible to assess the benefits of the 3D-data easily. We evaluate our approach with the publicly available Washington Dataset of real RGB-D data composed of 51 categories of household objects and obtained an improvement of around 10% in accuracy over the utilisation of 2D features. This result motivates further investigation when compared to some recently reported results tested on smaller datasets.**

***Index Terms*—Object Recognition, Convolutional Neural Networks, RGB-D data**

I. Introduction

Nowadays, there is great interest in understanding the way our brain processes 3D information for our everyday activities, amongst it, object recognition. The intricate system that underlies the process of visual recognition raises many questions about what type of processes are treated by different parts of the neural circuitry and how this complex processing is accomplished. Although the hierarchy between the areas that process the 3D information is not known precisely, successive theories in psychology [1], [2], [3], [4], [5] and neuroscience [6] have been tested, supported by advanced technologies of neural recording. In these works, it was discussed how the 3D information would be processed by the brain and if this processing would be directly related to object recognition tasks.

With the progress of 2.5D depth sensor and the rise of Convolutional Neural Networks as the most powerful approach for general recognition, studies have been conducted focusing on extracting 2D and 3D feature representations. But, although much progress was reported by these works [7], [8], [9], the recent emergence of approaches able to handle 3D-data in their pure form is a promising new direction [10], [11]. What these works have in common is the fact that a 3D representation in theory should have more information about the objects than a simpler 2D representation and should encompass details intrinsically related to the shape of the object which is not easily encoded in 2D images.

However, these approaches are only able to present better results when combined with multi-view solutions that process mainly 2D representations. Deep learning models support this viewpoint learning dependent theory since they achieve impressive results using a large collection of images (2D representation) [12], [13]. A system that achieves accuracy superior to human accuracy [14] was developed using the Imagenet dataset [12]. Neuroscience studies [6], although reinforcing the 3D processing as a fundamental part of the processing, do not neglect the multi-view importance to achieve this recognition.

With the aim of preserving the 3D spatial relationship between these components, as well as to integrate them, we propose an architecture based on Convolutional Neural Networks [15] extended from previous approaches [16], [17]. The proposed architecture was modelled and conceived to evaluate the use of 3D features by the visual system. The network extracts 3D features that are processed in all layers producing more complex 3D representations and at the same time preserving the spatial relationship between distinct components of the image. The generalization of the network is achieved through a multi-view training strategy.

This paper is organized in the following way: Section II discusses some of the approaches that are directly related to us, Section III presents our approach, detailing the neural network architecture, Section IV describes the experiments using a publicly available database of RGB-D images presented in [18] and in Section V, we conclude the paper and point out future directions.

This work was partially supported by CAPES Brazilian Federal Agency for the Support and Evaluation of Graduate Education (Process number 10441-13-1) and the European Union's Horizon 2020 Programme under HSC grant 642667 (SECURE).

In: Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), pages: 2068–2075, Rio de Janeiro, Brazil. IEEE, July, 2018.

II. Related Work

Some years ago, a large RGB-D dataset of household objects in multiple viewpoints was collected and used in experiments involving depth features, achieving improvements in recognition over just texture information [18]. During the later years, the work developed using this database was improved with several distinct approaches, mainly based on CNNs. What these approaches have in common is the fact that they handle the 2.5D data through depth maps using separate channels to extract appropriate features [18], [19], [7], [8], [9].

With the aim of developing an approach to learn hierarchical compositional part representations from raw 3D-data, Wu et al. [10] developed a Convolutional Deep Belief Network (3D *ShapeNet*) to model RGB-D data as a voxel grid. The model created can be used for recognition and also for shape completion. This work is different from previous approaches that are mainly based on depth maps and achieves good performance on a data set of synthetic object models using only shape information.

However, volumetric networks seem to be unable to capture all details of the objects [20], as the best reported results are related to multi-view approaches over 2D generalizations. An investigation conducted recently [11], [21], [22] lead the authors to propose two distinct architectures, with volumetric and multi-view specifications, with improvements over the previous approaches. Hua et al. [23] developed a point-wise convolution operation applied to every point in a point cloud, achieving comparable results. Recently, Caglayan and Can developed a network [24], [25] with competitive results employing single and multiple rotation recognition at testing time. Some of these results will be discussed in Section IV.

Our approach resembles these mentioned 3D networks [10], [11], [24] and it represents an extension of the 2.5D approaches. The CNN is composed of two separate channels, one to handle the 3D-data applying 3D convolution on the input (volumetric channel) and one to handle 2D data applying 2D convolution on images (standard channel). While the first one is responsible to preserve the structure of the data, the second is responsible for the flat recognition of the object. Both channels receive inputs representing the objects in multiple views during the training stage.

III. Neural Network Model

The neural architecture developed in our research receives as input a 3D representation of objects that allows us to retrieve the relative position of the points in relation to a reference frame located at the camera position. To accomplish this, we use RGB-D devices that provide, besides texture information, also depth information (x, y and z coordinates of each point). We also assume that the input to the neural network is a segmented object extracted from the entire scene.

We formalize an object that will be the input to our neural network as:

$$O = \{V_{\theta_1}, V_{\theta_2}, \dots, V_{\theta_n}\}\,, \tag{1}$$

where V is a set of viewpoints captured with camera orientation $\theta_i = (\text{roll}_i, \text{pitch}_i, \text{yaw}_i)$.

Every viewpoint V_{θ_i} represents the projection of all points $p_j = (x_j, y_j, z_j)$ captured in the x-y plane. In the case of 2D representations (RGB), this viewpoint is given by:

$$V^{\text{2D}}_{\theta_i} = \{p'_j \mid \forall p_j \in V_{\theta_i}\}\,, \tag{2}$$

where $p'_j = (x'_j, y'_j)$ is the projection of point p_j in the plane x-y.

If you consider the input to the neural network as a 3D input then each viewpoint is now defined as:

$$V^{\text{3D}}_{\theta_i} = \{P'_j \mid 1 \leq j \leq m\}\,, \tag{3}$$

where P'_j is a *slice* composed of projected points p'_j and m is the maximum number of slices. It is important to note that it is the parameter m that will specify the "resolution" of the 3D representation.

We consider that the size of each slice is given by

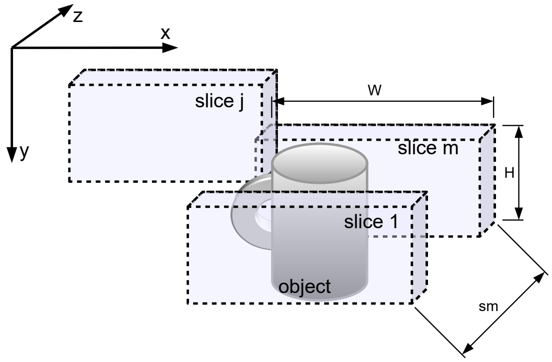
$$s = \frac{z_{\text{max}} - z_{\text{min}}}{m}\,, \tag{4}$$

where z_{min} and z_{max} are the maximal and minimum coordinate values of the points contained in the original point cloud (before projection). With this size computed, we can demonstrate easily that each slice contains points whose z coordinate is greater than $(j-1)s$ and smaller than js for $j \geq 1$.

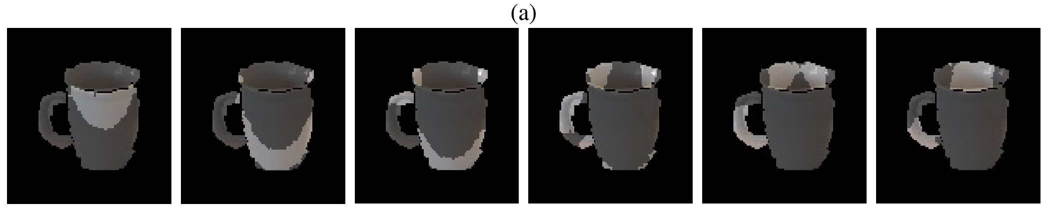
To better visualize this, Figure 1(a) shows a mug on a table with several slices composing the point cloud (only three slices are shown). The input captured in a given viewpoint $V^{\text{3D}}_{\theta_i}$ is then reshaped as a 4-dimensional matrix of size $m \times F \times W \times H$, where m is the number of slices, F is the set of features that characterizes the input (R, G and B channels for example), and W and H are the arbitrary horizontal and vertical sizes of the object's projection in the x-y plane. We can easily visualize this data as a cubic representation of dimension $W \times H \times m$ (the F would represent a fourth dimension, not shown in the figure). The Figures 1(b)-(g) show a real capture with the slices highlighted.

The neural network model employed here is a Convolutional Neural Network and the architecture is presented in Figure 2. There are two channels: the first channel deals with standard RGB images and the second channel deals with the depth information. Therefore, the architecture is handling 3D as well as 2D representations and can be trained using multi-view samples. The names `Object 2D` and `Object 3D` mean that the input to the CNN is one image with arbitrary $W \times H$ size and a set of slices with arbitrary dimension $W \times H \times m$, respectively. The tuples on the top of each object representation were instantiated according to the matrix formalization (m, F, H, W) for clarity. The size of the images decreases as the processing goes deeper as the result of the convolutional and pooling operations. For example, the dimension of the image decreases from 50×50 to 23×23 as the result of applying a convolutional filter of size $(5, 5)$ and a pooling filter of size $(2, 2)$. Details about the convolutional

In: Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), pages: 2068–2075, Rio de Janeiro, Brazil. IEEE, July, 2018.



(a)



(b) (c) (d) (e) (f) (g)

Fig. 1: (a) Object captured as a point cloud by a RGB-D device with reference frame x-y-z. The slices in the image divide the cloud into different parts that can be stacked together and reshaped as input to the CNN model. The slice j is floating just to represent its independence in relation to the others. (b)-(g) Real captures with some slices highlighted. The thickness of each slice was arbitrarily drawn for clarity.

and pooling operations and how to compute these values can be seen in previous works [15].

While the *Feature Maps Layer* is responsible for obtaining features that are increasingly complex and preserve the spatial relationships among them, the *Pooling Layer* is responsible to preserve the invariance of the features. Hubel and Wiesel [26] showed that the visual cortex is composed by cells that are sensitive to small regions named *receptive fields*. The receptive fields of these cells intersect with each other, allowing the convolutional operation to produce simple features, such as edges that will be grouped in subsequent layers into more complex features. In this way, the cells located in deeper layers can be seen as cells with larger receptive fields as well.

The 3D convolutional operation is defined as:

$$u_{fl}^{xyz} = tanh\left(b_{fl} + \sum_{n}^{f} \sum_{i}^{W^*} \sum_{j}^{H^*} \sum_{k}^{K^*} \phi\right) \quad (5)$$

and

$$\phi = w_{nl}^{ijk} u_{n(l-1)}^{(x+i)(y+j)(z+k)}, \quad (6)$$

where

- W^*, H^*, K^* represent the maximal dimensions of the receptive field used for the convolution. We are thus generalizing the concept of receptive field proposed by Hubel and Wiesel [26] to encompass 3D data. The dimensions W^* and H^* should not be confused with the dimensions W and H used for the object representation.
- u_{fl}^{xyz} is the activation cell of the (x, y, z) position in the feature map (or input) f in the layer l.
- w_{nl}^{ijk} is the weight in the receptive field position (i, j, k) at the feature map n in the layer l that is multiplied by the output activation value $u_{n(l-1)}^{(x+i)(y+j)(z+k)}$ in the layer $l-1$. Note that to obtain different values u_{fl}^{xyz}, the same set of w_{nl}^{ijk} is used. This means that, although the receptive field is activated by different regions of the image, distinct cells share the same weights. This is an important characteristic since this replication allows the generation of consistent 3D feature maps.

In a final stage, all the features obtained in both channels are transformed into a one-dimensional input vector and fed into a *Multilayer Perceptron* network with two hidden layers. The details of this implementation will be discussed in the next section.

IV. Results and Evaluation

For the experiments, we are using a publicly available dataset of RGB-D images [18]. This dataset is composed of approximately 42000 samples of objects grouped in 51 different categories. Each category is also subdivided in different instances. The database has 300 instances, such as several types of fruit, electronic devices (calculator, mobile phones, etc.), kitchen objects (bowls, mugs, etc.), vegetables, clothes, etc. The dataset provides a point cloud representation of the object as well as the equivalent RGB image. For each object, the samples were captured from multiple viewpoints and different heights relative to the ground, which allowed a multi-view evaluation.

In: Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), pages: 2068–2075, Rio de Janeiro, Brazil. IEEE, July, 2018.

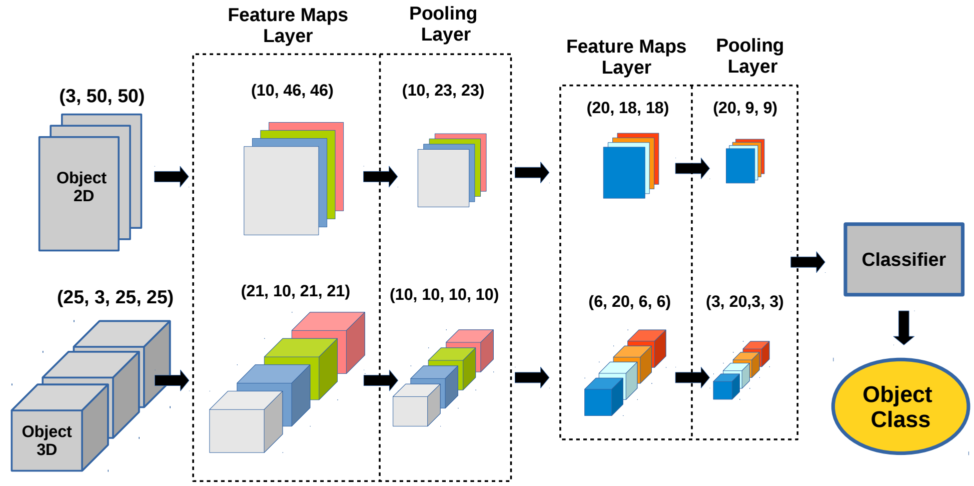


Fig. 2: The Convolutional Neural Network Architecture is composed of two channels, one for processing RBG images and one for processing RGB-D data. There are two convolutional layers, each followed by a pooling layer (max pooling). The convolutional layers extract features from the images, which are then combined in posterior layers. The max pooling layer is used for scale and translation invariance. The values used to describe the matrix (m, F, W, H) specify the number of slices, the number of features, and the arbitrarily chosen dimension $W \times H$ of the image. In the input, F is 3 to represent the RGB features. In this figure, the convolutional filters employed have sizes $(5, 5)$ for 2D objects and $(5, 5, 5)$ for 3D objects. The pooling filters employed have sizes $(2, 2)$ for 2D objects and $(2, 2, 2)$ for 3D objects.

TABLE I: Parameters used in the experiments

Parameter	Training-validation-testing Experiment	Leave-one-instance-out Experiment
Image size channel 1	$(50, 50)$	$(50, 50)$
Image size channel 2	$(25, 25)$	$(25, 25)$
Convolutional filter size	$(5, 5)$ / $(5, 5)$	$(5, 5)$ / $(5, 5)$
Pooling filter size	$(2, 2)$ / $(2, 2)$	$(2, 2)$ / $(2, 2)$
Number of features maps	10 / 20	10 / 20
Learning rate	0.01	0.005
Momentum	0.09	0.09
Batchsize	30	30
Neurons in hidden layer	250	250
Number of epochs	100	100
Training-validation-testing size	$(60\%, 20\%, 20\%)$	–
Number of categories used	51	25

TABLE II: Accuracy of the methods

	RGB	RGB+D	RGB+RGBD
Leave-one-instance-out Experiment	$67, 67\% \pm 5, 76$	$75, 86\% \pm 6, 43$	$76, 38\% \pm 6, 20$
Training-validation-testing Experiment	$77, 73\% \pm 3, 15$	$81, 29\% \pm 9, 25$	$86, 82\% \pm 3, 83$

The neural network is implemented in Theano and run on a GPU Nvidia. We use backpropagation and stochastic gradient descent for training the network. The input to our network is extracted directly from the point cloud samples. To generate the images that feed the 2D channel, each point cloud is projected into the x-y plane, with $W = 300$ and $H = 300$, but retaining the aspect ratio of the object. The width and height are redimensioned to smaller sizes before using the image as input to the CNN. For the 3D case the same procedure is assumed and we also defined the number of slices m as 25. This was empirically determined and motivated by efficiency reasons. Table I shows the parameters used in the experiments. We started with the same parameters used in our previous work [17]. Henceforth, we performed a systematic search in the parameter space and defined the values that performed better. We dedicated special attention to the number of feature maps, number of channels and number of layers as these parameters are strictly related to the CNN performance.

In the next paragraphs, the results of two distinct experiments will be detailed: 1) Leave-one-instance-out and 2) Training-validation-testing. Each of these are tested over 3 different CNNs: 1) RGB network composed of 1 channel for which the input is an RGB image (i.e, $m = 1$), 2) RGB+D network composed of 2 channels for which the input is an RGB

In: Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), pages: 2068–2075, Rio de Janeiro, Brazil. IEEE, July, 2018.

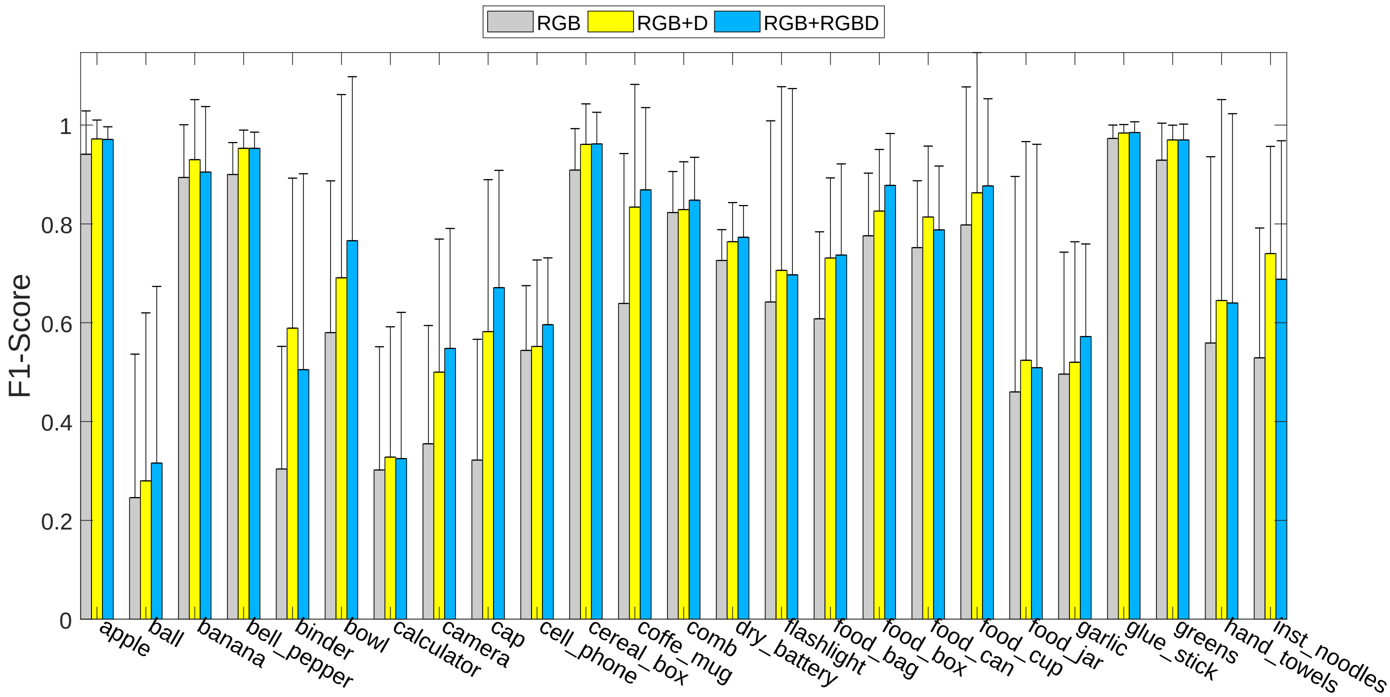


Fig. 3: F1-scores for each of the 25 objects in the Leave-one-instance-out experiment. The use of depth information improved the accuracy in the 25 cases tested. The standard deviation was relatively large since each run left one random and distinct object out (that could be an instance more difficult for the network to generalize).

image and an RGB+D representation (i.e, m=25), with no color information in the second channel and 3) RGB+RGBD network composed of 2 channels for which the input is an RGB image and an RGB+RGBD representation (i.e, m=25), but now with color information in the second channel. The results can be seen in Table II.

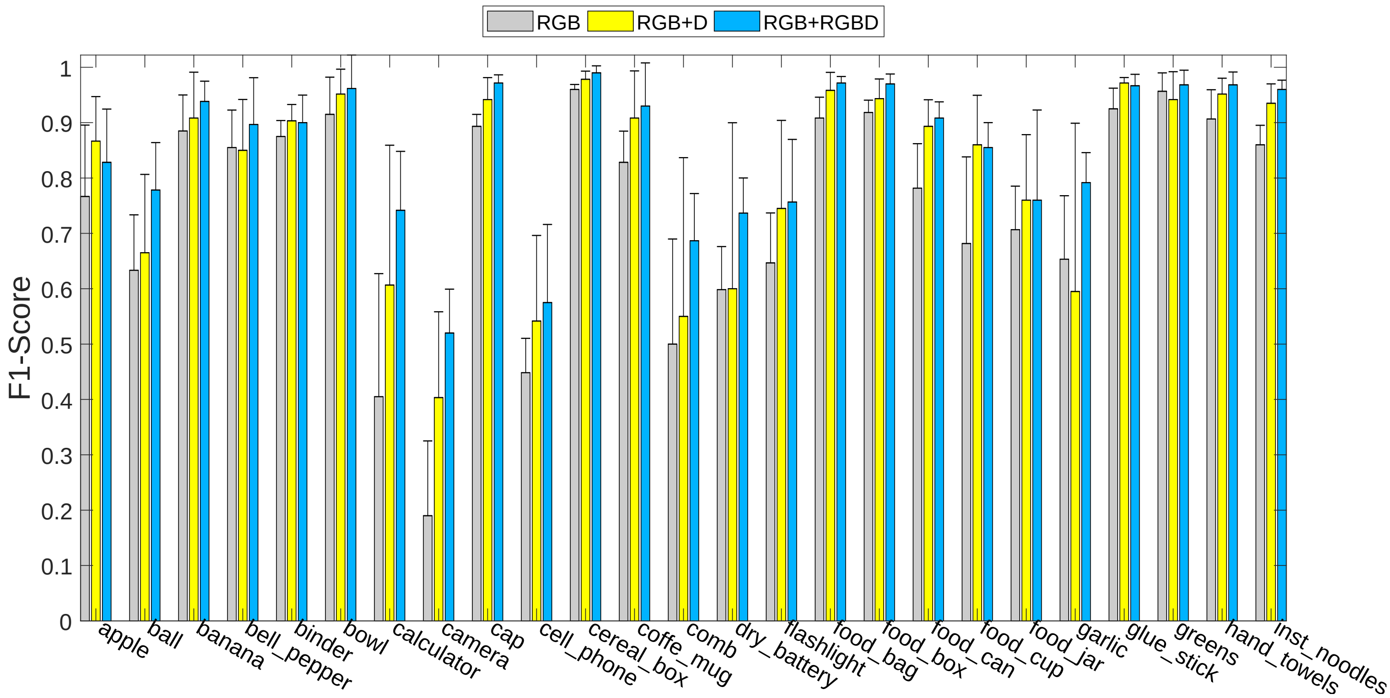
There are few differences in the value of the parameters used in both experiments: besides the number of categories employed, the learning rate in the Leave-one-instance-out experiment was smaller since this value demonstrated to be more stable during training, presenting less instability in the error computed. We also would like to point out the relatively small number of features employed in each layer. The motivation was to build simplified architectures to evaluate if the proposed method would have a good performance even for simple networks.

The Leave-one-instance-out experiment selects randomly one instance per object to be left out of the training set (these instances are then used only in the test stage), according to the cross-validation approach used [18]. We were able to test with a smaller subset of 25 categories due to performance constraints. Therefore, the proposed CNN cannot be directly compared with other approaches that used all the 51 categories. For example, Caglayan and Can [24] reported an accuracy of 82.4 ± 2.2, which is better than our Leave-one-instance-out experiment but worse than our Training-validation-testing experiment. We plan to make our experimental setup comparable in future works.

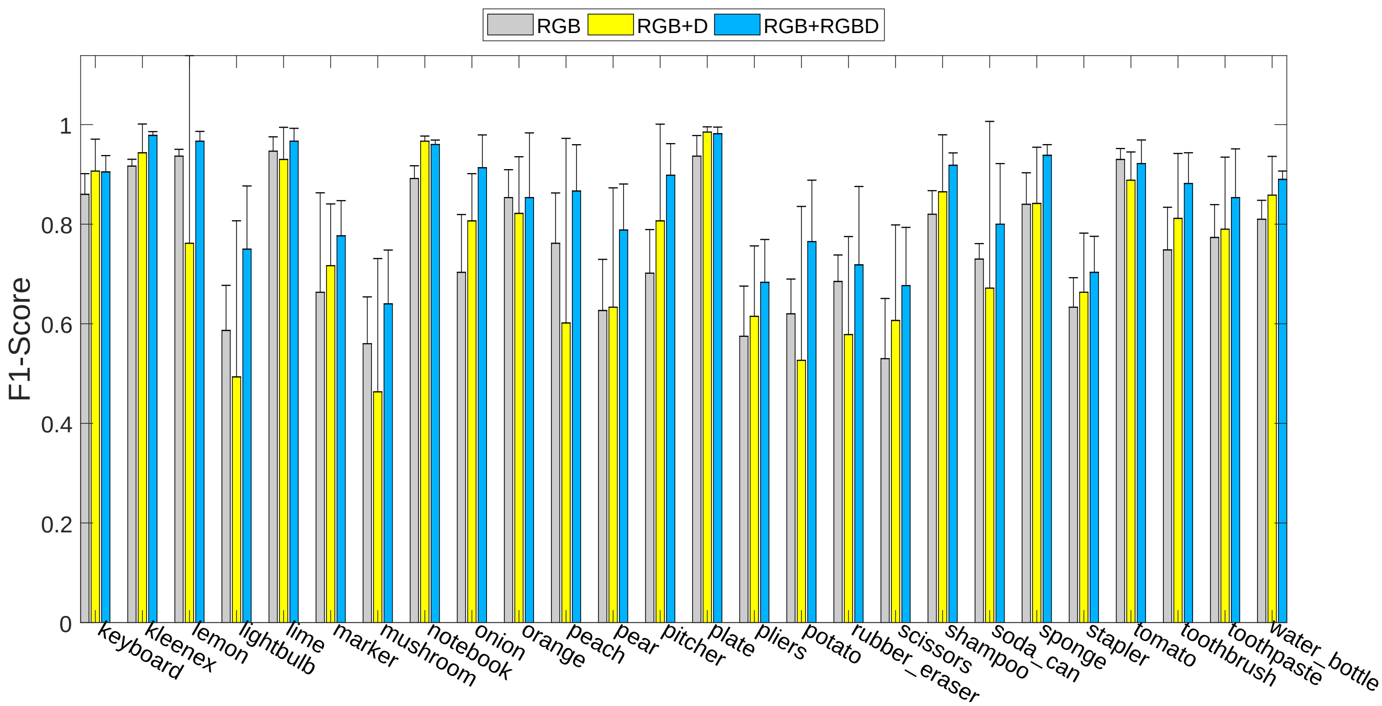
We can see the improvement obtained using depth information instead of just texture information. Besides that, the standard deviation is larger than 5% in the three cases reported. The reason for that can be better understood in Figure 3. As can be seen, some objects have relatively large standard deviations due to the random selection of the instance chosen to be left out. As these instances are not the same for every run, in some cases the generalisation of the network is not so effective. The accuracies and standard deviations reported for the RGB+D and RGB+RGBD network are very similar. Looking at Figure 3, we can see that sometimes the RGB+D network performed better (i.e, inst_noodles and binder), while in other cases the RGB+RGBD performed better (i.e, bowl and cap), but overall they both present very similar results. This indicates that the color information is not so valuable for the generalisation of instances not previously seen, as expected. For these instances, shape information is more relevant. This result is consistent if we consider that instances can have distinct textures that at first should not be a determinant factor to categorize an object.

The full dataset is used in the Training-validation-testing experiment. In this case, from the samples, 60% are selected for training, 20% for validation and 20% for testing. The value of 60% is chosen to have fewer training samples than the number of training samples of the Leave-one-instance-out experiment. In that case, only one instance per category was left out, which means that a rate of approximately 80% was used for training. Thus, as we are providing random samples

In: Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), pages: 2068–2075, Rio de Janeiro, Brazil. IEEE, July, 2018.



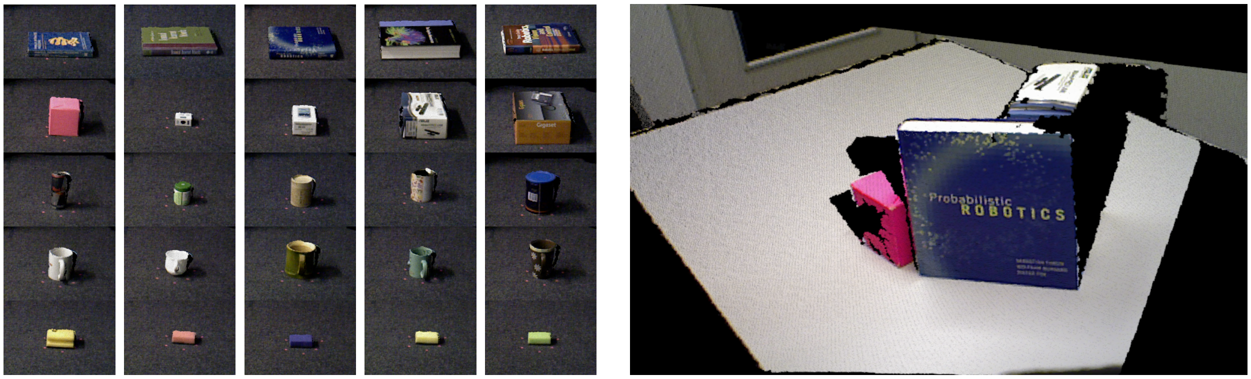
(a)



(b)

Fig. 4: F1-scores for each of the 51 objects in the Training-validation-testing experiment. The use of depth information improved the accuracy in all cases for all objects. (a) shows objects 1 to 25, (b) shows objects 26 to 51.

In: Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), pages: 2068–2075, Rio de Janeiro, Brazil. IEEE, July, 2018.



(a) (b)

Fig. 5: (a) Objects used in the experiment. (b) Scenario used in the multi-view test.

from all instances, we can decrease the number of training samples. The result obtained is the average of 6 trials and can be seen also in Table II, where the use of depth again improves the accuracy reported. The F1-scores for all 51 objects can be seen in Figures 4(a) (objects 1 to 25) and 4(b) (objects 26 to 51). The results are divided in two figures for better visualization, but they are obtained for all categories together.

A similar behaviour to the last experiment is observed for the F1-scores, with the RGB+D and RGB+RGBD networks performing better than the RGB network for all objects. As we are now providing random samples from all instances as input to the network, the color information demonstrated to be an important factor to distinguish categories. This fact can be checked observing that the RGB+D network rarely performs better than the RGB+RGBD network and when this occurs, the difference is not large (i.e, apple and banana).

V. Conclusion and Future Research

We developed a Convolutional Neural Network composed of two channels that extracts 3D features from the objects provided as input. We found that the 3D features improved the accuracy of object categorization over the utilisation of texture information by approximately 10%.

In experiments in which some objects were left out of the training stage, the network presented a similar accuracy for 3D features with color and without color information. This indicates that, for new objects not previously seen, the color information was not an important factor, and the generalisation to the correct category was based mainly on the shape of the object. On the other hand, for the experiments involving random samples collected from all instances, the use of color in the 3D features was important, since similar objects in size and shape, that could have been presented as input to the neural

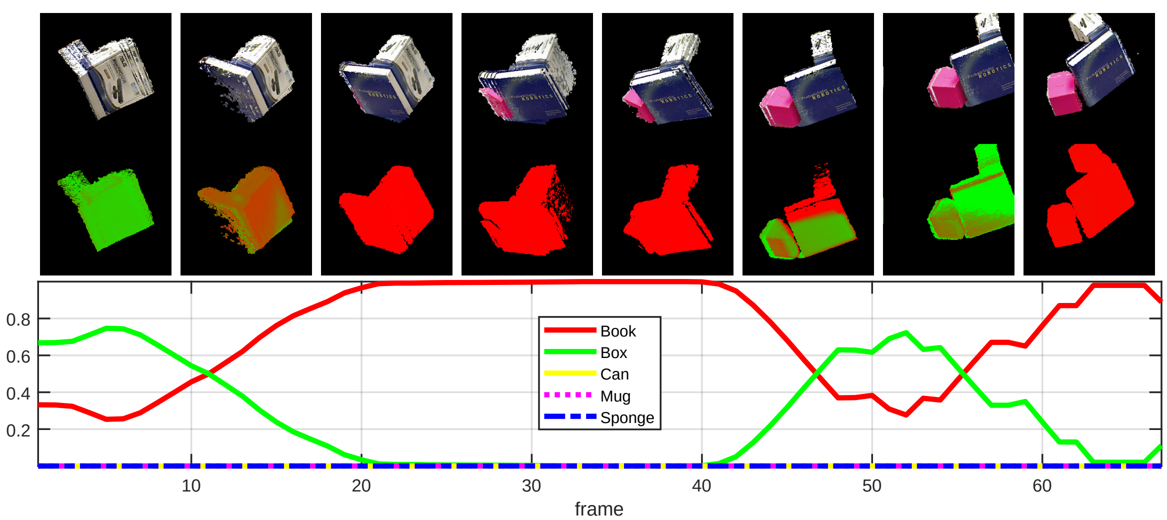


Fig. 6: The top row represents the raw capture of 3 objects from the RGB-D device ($\approx$ 70 frames). This sequence of captures was performed rotating the RGB-D device clockwise. The lower row represents the categorization performed by the network. The color indicates the probability of each point to belong to one of the 5 classes. The neural activation is shown in the lower row and we can see that the classes "Box" and "Book" are activated mostly, while other classes are not being activated.

network at the training stage, can be differentiated only by texture information.

We also did preliminary multi-view experiments based on recognition performed over time. The recognition was performed with the RGB-D device moving from the right to the left around a table. We capture a small database composed of 25 objects from 5 five different categories: book, box, can, mug and sponge (Figure 5). Each object was captured in 6 different views from a specific distance and height in relation to the camera. The total amount of images captured was then $25 \times 6 = 150$. To increase the number of samples as input to our neural network, we rotated each point cloud in 50 distinct orientations of $roll$, $pitch$ and yaw, therefore obtaining $150 \times 50 + 150 = 7650$ samples.

In Figure 6, the network starts with the largest neural activity related to the box, as this is the main object in the viewpoint of the camera. Around frame 15, the book starts to dominate the output of the neural network, and this situation is maintained until frame 40 when features related to the second box appear in the capture. From this point, the behaviour of the neural network is alternating between the box and the book and this can be noted by the color of the objects indicating the probabilities to belong to one class or another. It seems reasonable to think that the objects located in the central position of the image subdue the other objects. It is also possible to see that the other objects (can, mug and sponge), which are not present in the image, have no activity in the neural network output.

As the objects are close to each other, the neural network is not able to clearly separate the objects because the input is provided as one block of indistinct features. The concept behind this procedure was to provide a way to label the objects with several classes. The color of the object around frame 50 indicates a huge dominance of the class box, but many regions of the object are still classified as a book. We plan to work on a formal modelling of this problem as well as improve the experimental setup.

Finally, we also plan to compare our 3D CNN approach with other recent approaches [10], [11], [24], adapting our Leaving-one-object-out cross-validation experiment to draw comparable results.

REFERENCES

[1] D. Marr and H. K. Nishihara. Representation and recognition of the spatial organization of three-dimensional structure. *Proceedings of the Royal Society of London B: Biological Sciences*, 200(1140):269–294, 1978.

[2] S. Ullman. Aligning pictorial descriptions: An approach to object recognition . *Cognition*, 32(3):193 – 254, 1989.

[3] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[4] H. H. Bülthoff, S. Y. Edelman, and M. J. Tarr. How are three-dimensional objects represented in the brain? *Cerebral Cortex*, 5(3):247–260, 5 1995.

[5] M. J. Tarr and S. Pinker. Mental rotation and orientation-dependence in shape recognition. *Cognitive Psychology*, 21(2):233 – 282, 1989.

[6] Y. Yamane, E. T. Carlson, K. C. Bowman, Z. Wang, and C. E. Connor. A neural code for three-dimensional object shape in macaque inferotemporal cortex. *Nature Neuroscience*, 11:1352–1360, 2008.

[7] R. Socher, B. Huval, B. Bath, C. D Manning, and Andrew Y. Ng. Convolutional-recursive deep learning for 3D object classification. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 656–664. Curran Associates, Inc., 2012.

[8] A. Eitel, J. T. Springenberg, L. Spinello, M. A. Riedmiller, and W. Burgard. Multimodal deep learning for robust RGB-D object recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 681–687, 2015.

[9] M. Schwarz, H. Schulz, and S. Behnke. RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. In *IEEE International Conference on Robotics and Automation, ICRA, Seattle, WA, USA*, pages 1329–1335, 2015.

[10] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920. IEEE Computer Society, 2015.

[11] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. Volumetric and multi-view CNNs for object classification on 3D data. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.

[12] Stanford Vision Lab. Large scale visual recognition challenge (ILSVRC).

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[16] P. Barros, S. Magg, C. Weber, and S. Wermter. A Multichannel Convolutional Neural Network for Hand Posture Recognition. In *International Conference on Artificial Neural Networks*, pages 403–410, 2014.

[17] M. B. Soares, P. Barros, G. I. Parisi, and S. Wermter. Learning Objects from RGB-D Sensors Using Point Cloud-based Neural Networks. In *European Symposium on Artificial Neural Networks (ESANN'2015)*, pages 439–444, 2015.

[18] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *ICRA*, pages 1817–1824. IEEE, 2011.

[19] L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for RGB-D based object recognition. In *In International Symposium on Experimental Robotics (ISER*, 2012.

[20] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 945–953, Washington, DC, USA, 2015. IEEE Computer Society.

[21] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 77–85, 2017.

[22] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum PointNets for 3D object detection from RGB-D data. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2018.

[23] B.-S. Hua, M.-K. Tran, and S.-K. Yeung. Point-wise Convolutional Neural Network. *ArXiv e-prints*, 2017.

[24] A. Caglayan and A. Can. Volumetric object recognition using 3D CNNs on depth data. PP:1–1, 03 2018.

[25] A. Caglayan and A. B. Can. 3D convolutional object recognition using volumetric representations of depth data. In *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pages 125–128, 2017.

[26] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology (London)*, 195:215–243, 1968.