# Recurrent Convolutional Fusion for RGB-D Object Recognition

Mohammad Reza Loghmani, Mirco Planamente, Barbara Caputo, and Markus Vincze

***Abstract*—Providing robots with the ability to recognize objects like humans has always been one of the primary goals of robot vision. The introduction of RGB-D cameras has paved the way for a significant leap forward in this direction thanks to the rich information provided by these sensors. However, the robot vision community still lacks an effective method to synergically use the RGB and depth data to improve object recognition. In order to take a step in this direction, we introduce a novel end-to-end architecture for RGB-D object recognition called recurrent convolutional fusion (RCFusion). Our method generates compact and highly discriminative multi-modal features by combining RGB and depth information representing different levels of abstraction. Extensive experiments on two popular datasets, RGB-D Object Dataset and JHUIT-50, show that RCFusion significantly outperforms state-of-the-art approaches in both the object categorization and instance recognition tasks. In addition, experiments on the more challenging Object Clutter Indoor Dataset confirm the validity of our method in the presence of clutter and occlusion. The code is publicly available at: "https://github.com/MRLoghmani/rcfusion."**

***Index Terms*—RGB-D perception, recognition, visual learning.**

## I. Introduction

HUMAN-BUILT environments are, ultimately, collections of objects. Every daily activity requires to understand and operate a set of objects to accomplish a task. Robotic systems that aim at assisting the user in his own environment need to possess the ability to recognize objects. In fact, object recognition is the foundation for higher-level tasks that rely on an accurate description of the visual scene.

Despite the interesting results achieved for object recognition using standard RGB images, there are inherent limitations due to the loss of data caused by projecting the 3-dimensional world into a 2-dimensional image plane. The use of RGB-D (Kinect-style) cameras alleviates these shortcomings by using range imaging technologies to provide information about the camera-scene distance as a depth image. These sensors became ubiquitous in robotics due to their affordable price and the rich visual information they provide. In fact, while the RGB image contains color, texture and appearance information, the depth image contains additional geometric information and is more robust with respect to lighting and color variations. Since RGB-D cameras are already deployed in most service robots, improving the performance of robot perceptual systems through a better integration of RGB and depth information constitutes a "free lunch".

After the pivotal work of Krizhevsky *et al.* [1], deep convolutional neural networks (CNNs) quickly became the dominant tool in computer vision, establishing new state-of-the-art results for a large variety of tasks. Research in RGB-D object recognition followed the same trend, with numerous algorithms (e.g. [2]–[4]) exploiting features learned from CNNs instead of the traditional hand-crafted features. The common pipeline involves two CNN streams, operating on RGB and depth images respectively, as feature extractors. However, the lack of a large-scale dataset of depth images to train the depth CNN forced the vision community to find practical workarounds. Much effort has been dedicated to develop methods that colorize the depth images to exploit CNNs pre-trained on RGB images. However, the actual strategies to extract and combine the features from the two modalities have been neglected. Several methods simply extract features from a specific layer of the two CNNs and combine them through a fully connected or a max pooling layer. We argue that these strategies are sub-optimal because (a) they assume that the selected layer always represents the best abstraction level to combine RGB and depth information and (b) they do not exploit the full range of information from the two modalities during the fusion process.

In this letter, we propose a novel end-to-end architecture for RGB-D object recognition called recurrent convolutional fusion (RCFusion). Our method extracts features from multiple hidden layers of the CNNs for RGB and depth, respectively, and combines them through a recurrent neural network (RNN), as shown in Figure 1. Our idea is that combining RGB and depth features from several levels of abstraction can provide greater information to the classifier to make the final prediction. Although RNNs are typically used to process sequential data, this type of neural networks have been proven to be very effective information compressors [5] and scale well in the parameters with respect to the number of extracted features, as discussed in Section III-C. In addition, we provide experimental evidence that this solution is superior to simply fusing the concatenated

Manuscript received February 14, 2019; accepted May 29, 2019. Date of publication June 6, 2019; date of current version June 24, 2019. This letter was recommended for publication by Associate Editor T. Pham and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported in part by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Grant agreement 676157, project ACROSSING, and in part by the ERC Grant 637076—RoboExNovo (B.C.) and the CHIST-ERA project ALOOF (B.C.) *(Corresponding author: Mohammad Reza Loghmani.)*

M. R. Loghmani and M. Vincze are with the Vision4Robotics Group, Automation and Control Institute, TU Wien, 1040 Vienna, Austria (e-mail: loghmani@acin.tuwien.ac.at; vincze@acin.tuwien.ac.at).

M. Planamente and B. Caputo are with the VANDAL Laboratory, Italian Institute of Technology, 10144 Milan, Italy (e-mail: Mirco.Planamente@iit.it; Barbara.Caputo@iit.it).

Digital Object Identifier 10.1109/LRA.2019.2921506

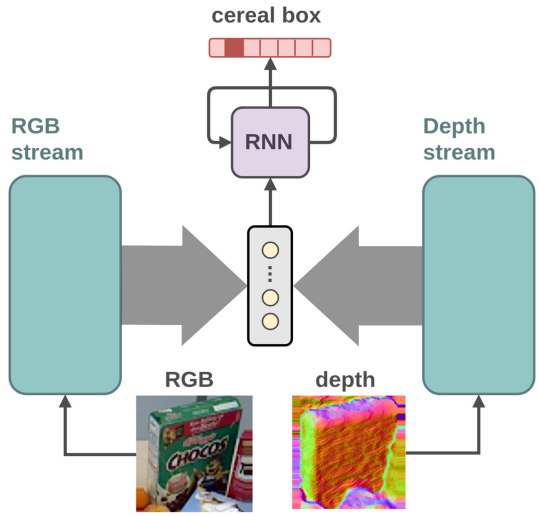


Fig. 1. High-level scheme of RCFusion. The blue boxes are instantiated with convolutional neural networks and the thick arrows represent multiple feature vectors extracted from different layers of a CNN.

features with a fully connected layer (see ablation study in Section IV-D).

We evaluate our method on standard object recognition benchmarks, RGB-D Object Dataset [6] and JHUIT-50 [7], and we compare the results with the best performing methods in the literature. The experimental results show that our method outperforms the existing approaches and establishes new state-of-the-art results for both datasets. In order to further consolidate the effectiveness of our method, we adapt an object segmentation dataset, called Object Clutter Indoor Dataset (OCID) [8], to the instance recognition task to further evaluate RCFusion. OCID has been recently released to provide object scenes with high level of clutter and occlusion, arguably two of the biggest challenges faced by robotic visual perception systems [9]. Our method confirms its efficacy also on this challenging dataset, despite the small amount of training data available. An implementation of the method, relying on tensorflow [10], is publicly available at: https://github.com/MRLoghmani/rcfusion.

In summary, our contributions are:

- a novel architecture for RGB-D object recognition that sequentially combines RGB and depth features representing different levels of abstraction,
- state-of-the-art performance on the most popular RGB-D object recognition benchmark datasets,
- introduction of a new benchmark with robotic-oriented challenges, i.e. clutter, occlusion and little training data.

The remainder of the letter is organized as follows: the next section positions our approach compared to related work, Section III introduces the proposed method, Section IV presents the experimental results and Section V draws the conclusions.

## II. Related Work

The diffusion of RGB-D cameras fueled an increasing effort in designing visual algorithms able to exploit the additional depth information provided by these sensors. Classical approaches for RGB-D object recognition (e.g. [6], [11]) used a combination of different hand-crafted feature descriptors, such as SIFT, textons, and depth edges, on the two modalities (RGB and depth) to perform object matching. More recently, several methods have exploited shallow learning techniques to generate features from RGB-D data in an unsupervised learning framework [12]–[14].

Since the ground-breaking work of Krizhevsky *et al.* [1], data-hungry deep CNNs have been the go-to solution for feature extraction. While large-scale datasets of RGB images, such as ImageNet [15], allowed the generation of powerful CNN-based models for RGB feature extraction, the lack of a depth counterpart posed the problem of how to extract features from depth images. An effective and convenient strategy to circumvent the problem is to colorize the depth images to exploit CNNs pre-trained on RGB data. Several hand-crafted colorization approaches have been proposed to map the raw depth value of each pixel [2] or derived physical quantities, such as position and orientation [16] or local surface normals [17], to colors. Carlucci *et al.* [18] proposed instead a leaning-based approach to colorize the depth images by training a colorization network. Other methods use alternatives to RGB-trained CNNs for extracting features from depth data. Li *et al.* [19] generate the depth features using a modified version of HONV [20] encoded with Fisher Vector [21]. Carlucci *et al.* [4] generate artificial depth data using 3D CAD models to train a CNN that extracts features directly from raw depth images.

The aforementioned methods focus on effectively extracting features from the depth data and use trivial strategies to combine the two modalities for the final prediction. For example, Carlucci *et al.* [18] simply select the class with the highest activation among the RGB and depth predictions, while Eitel *et al.* [2] and Aakerberg *et al.* [3] use a fully connected layer to learn to fuse the predictions from the two modalities. Alternatively, a few works prioritize the development of an effective modality fusion. Wang *et al.* [22] alternate between maximizing the discriminative characteristics of each modality and minimizing the inter-modality distance in feature space. Wang *et al.* [23] obtain the multi-modal feature by using a custom layer to separate the individual and correlated information of the extracted RGB and depth features. Both methods combine the two modalities by processing features extracted from one layer of the CNNs and rely on cumbersome multi-stage optimization processes.

Recent works from related areas, such as object detection and segmentation from color images, show the benefits of using features extracted from multiple layers of a CNN. Hariharan *et al.* [24], increase the resolution of higher level features by combining information from lower layers at a pixel level for segmentation purposes. Bell *et al.* [25] perform object detection at different scales using features extracted from different layers of a pre-trained network. These methods mostly take advantage of the difference in receptive fields in various layers of the neural network and use a simple combination of pooling and linear transformations to process the extracted features.

The focus of this letter is on the synthesis of multi-modal features from RGB-D data rather than the depth processing. In fact, for the depth processing part, we adopt the standard colorization method based on surface normals, since it has been proven not only to be the most effective non-learned colorization method

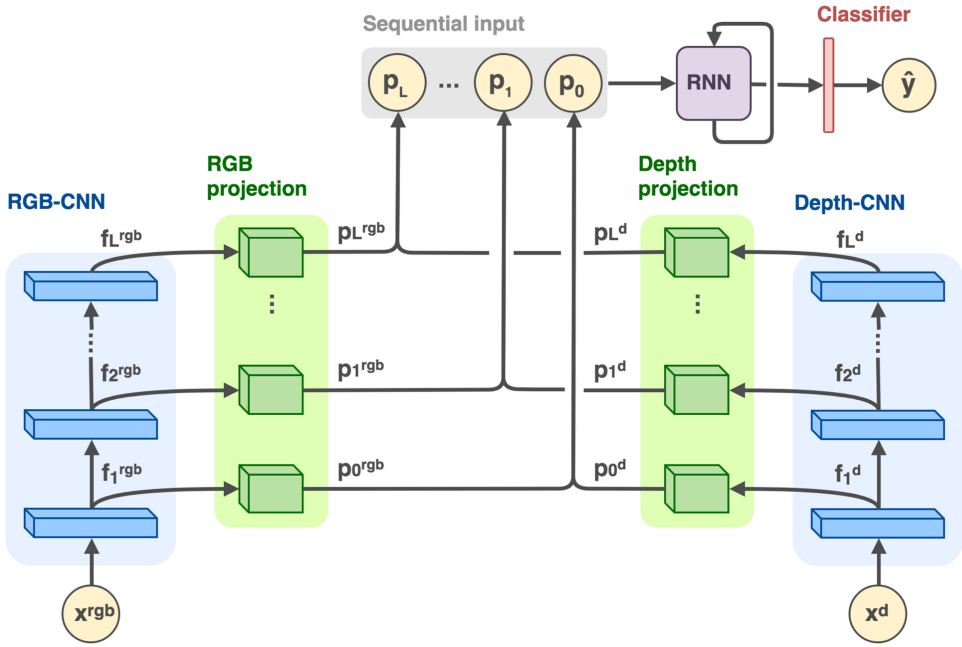


Fig. 2. Architecture of recurrent convolutional fusion. It consists of two streams of convolutional neural networks (CNN) that process RGB and depth images, respectively. The output of corresponding hidden layers from the two streams are projected into a common space, concatenated and sequentially fed into a recurrent neural network (RNN) that synthesizes the final multi-modal features. The output of the RNN is then used by a classifier to determine the label of the input data.

in the literature, but also to surpass learned methods in some instances [18]. Differently from existing works, our method produces highly informative global features from different levels of abstraction through a dedicated non-linear unit, called projection block. Features from the RGB and depth modalities are then combined together in a sequential manner to generate the final multi-modal representation. In addition, our model can be trained end-to-end, without the need of optimizing in multiple stages.

## III. Recurrent Convolutional Fusion

Our multi-modal deep neural network for RGB-D object recognition is illustrated in Figure 2. The network's architecture has three main stages:

1) *Multi-level feature extraction:* two streams of convolutional networks, with the same architecture, are used to process RGB and depth data (RGB-CNN and Depth-CNN), respectively, and extract features at different levels of the networks;
2) *Feature projection and concatenation:* features extracted from each level of the RGB- and Depth-CNN are individually transformed through projection blocks and concatenated to create the corresponding RGB-D feature;
3) *Recurrent multi-modal fusion:* RGB-D features extracted from different levels are sequentially fed to an RNN that produces a descriptive and compact multi-modal feature.

The output of the recurrent network is then used by a softmax classifier to infer the object label. The network can be trained end-to-end with a cross-entropy loss using standard backpropagation algorithms based on stochastic gradient descent. In the following, we describe in greater detail each of the aforestated characteristics of RCFusion.

### A. Multi-Level Feature Extraction

CNNs process the input with sets of filters learned from a large amount of data. These filters represent progressively higher levels of abstraction, going from the input to the output: edges, textures, patterns, parts, and objects [26]. Methods for RGB-D object recognition commonly combine the output of one of the last layers of the RGB- and Depth-CNN (typically the last layer before the classifier) and assume that the chosen layer represents the appropriate level of abstraction to combine the two modalities. We argue that it is possible to remove this assumption by combining RGB and depth information at multiple layers across the CNNs and use them all to generate a highly discriminative RGB-D feature. Let us denote with $x^{rgb} \in \mathcal{X}^{rgb}$ the RGB input images, with $x^d \in \mathcal{X}^d$ the depth input images and $y \in \mathcal{Y}$ the labels, where $\mathcal{X}^{rgb}$, $\mathcal{X}^d$ and $\mathcal{Y}$ are the RGB/depth input and label space. We further denote with $f_i^{rgb}$ and $f_i^d$ the output of layer $i$ of RGB-CNN and Depth-CNN, respectively, with $i = 1, ..., L$ and $L$ the total number of layers of each CNN. Notably, visualizing the learned filters has shown [26] that, for a given task, a chosen CNN architecture consistently generates features with the same level of abstraction from a reference layer. For example, AlexNet [1] learns various types of Gabor filters in the first convolutional layer. So, the same architecture is chosen for RGB- and Depth-CNN to ensure the same abstraction level at corresponding layers.

### B. Feature Projection and Concatenation

One of the main challenges in combining features obtained from different hidden layers of the same network is the lack of a one-to-one correspondence between elements of the different feature vectors. More formally, $f_i^*$ and $f_j^*$, with $i \neq j$ and $*$ indicating any of the superscripts $rgb$ or $d$, have (in general)

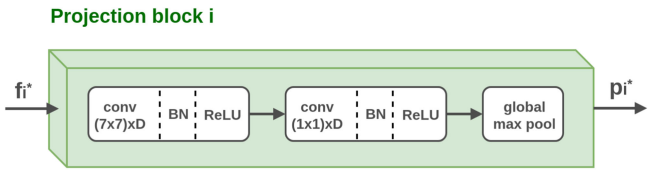


Fig. 3. Implementation of the projection block that transforms the feature $f_i^*$ into the projected feature $p_i^*$. $conv(k \times k) \times D$ indicates a convolutional layer with $D$ filters of size $(k \times k)$, $BN$ indicates a batch normalization layer and $ReLU$ indicates an activation layer with ReLU non-linearity.

different dimensions and thus belong to distinct feature spaces, $\mathcal{F}_i$ and $\mathcal{F}_j$. In order to make features coming from different levels of abstraction comparable, we project them into a common space $\bar{\mathcal{F}}$:

$$p_i^* = G_i^*(f_i^*) \quad \text{s.t.} \quad p_i^* \in \bar{\mathcal{F}} \tag{1}$$

The projection block $G_i(.)$ performs a set of non-linear operations to transform a volumetric input into a vector of dimensions (1×D). More specifically, $G_i(.)$ is defined by two convolutional layers (with batch normalization and ReLU non-linearity) and a global max pooling layer, as shown in Figure 3. The projected RGB and depth features of each layer $i$ are then concatenated to form $p_i = \left[p_i^{rgb}; p_i^d\right]$.

### C. Recurrent Multi-Modal Fusion

In order to create a compact multi-modal representation, the set $\{p_1, \ldots, p_L\}$ is sequentially fed to an RNN. Recurrent models align the positions of the elements in the sequence to steps in computation time and generate a sequence of hidden states $h_i$ as a function of the previous hidden state $h_{i-1}$ and the current input $p_i$. In this letter, we use an instantiation of an RNN called gated recurrent unit (GRU) [27]. This network is considered to be a variation of long-short term memory (LSTM) [28] that requires $25\%$ less parameters. GRU has been proven to be able to retain information even in extremely long sequences with thousands of elements [5].

GRU computes the $n^{th}$ element of the hidden state at step $i$ as an adaptive linear interpolation:

$$h_i^n = (1 - z_i^n)h_{i-1}^n + z_i^n \tilde{h}_i^n, \tag{2}$$

where $z_i^n$ is called update gate and is computed as

$$z_i^n = sigmoid(\theta_z p_i + \gamma_z h_i)^n, \tag{3}$$

where $sigmoid(.)$ is the sigmoid function and $\theta_z$ and $\gamma_z$ are the trainable parameters of the gate. Essentially, the update gate determines how much the unit updates its content. The candidate activation $\tilde{h}_i$ in Equation 2 is computed as

$$\tilde{h}_i^n = tanh(\theta_h p_i + \gamma_h(r_i \odot h_{i-1}))^n, \tag{4}$$

where $r_i$ is the reset gate, $\theta_h$ and $\gamma_h$ are trainable parameters and $\odot$ is the element-wise multiplication operation. Similarly to $z_i^n$, the reset gate $r_i^n$ is computed as

$$r_i^n = sigmoid(\theta_r p_i + \gamma_r h_i)^n, \tag{5}$$

where $\theta_r$ and $\gamma_r$ are the trainable parameters of the gate. When $r_i^n$ assumes values close to zero, it effectively resets the hidden state of the network to the current input $p_i$. This double-gate mechanism has the goal of ensuring that the hidden state progressively embeds the most relevant information of the input sequence $\{p_1, \ldots, p_L\}$.

The RNN, combined with a softmax classifier, models a probability distribution over a sequence by being trained to predict the category label given the sequence of projected RGB-D features. In particular, the prediction of the $j^{th}$ class of the multinomial distribution of $K$ object categories is obtained as

$$\hat{y}_j = Pr(y_j = 1|p_1, ..., p_1) = \frac{exp(h_L^T \theta_c^j)}{\sum_{k=1}^{K} exp(h_L^T \theta_c^k)}, \tag{6}$$

where $\theta$ is the matrix of trainable parameters of the classifier and $\theta_{j(/k)}$ represents its $j^{th}(/k^{th})$ row, and the superscript $T$ represents the transpose operation.

The choice of a recurrent network for this operation is twofold: (a) the hidden state of the network acts as a memory unit and embeds a summary of the most relevant information from the different levels of abstraction, and (b) the number of parameters of the network is independent of $L$, while for a more straightforward choice, such as a fully connected layer, it grows linearly with $L$. Although RNNs are typically used to process time series data, our atypical deployment is supported by previous works [29], [30] that have shown that these type of networks are also useful in compressing and combining information from different sources. We empirically demonstrate in the ablation study in Section IV-D that a recurrent network is more effectively than a typical fully connected layer to aggregate the RGB and depth features from different levels of abstraction.

## IV. Experiments

In the following, we evaluate RCFusion on RGB-D Object Dataset, JHUIT-50, and OCID. After revealing the protocol used to set up the experiments, we discuss the setting used for training the network. Then, we show how the performances of our method compare to the existing literature. Finally, we perform an ablation study to identify the contribution of the different elements of our method.

### A. Datasets

**RGB-D Object Dataset:** It contains 41,877 RGB-D images capturing 300 objects from 51 categories, spanning from fruit and vegetables to tools and containers. Since its introduction, this dataset has become the silver thread connecting the existing methods for RGB-D object recognition. We use this dataset to assess the performance of RCFusion in the object categorization task. For the evaluation, we follow the standard experimental protocol defined in [11], where ten training/test split are defined in such a way that one object instance per class is left out of the training set. The reported results are the average accuracy over the different splits.

**JHUIT-50:** It contains 14,698 RGB-D images capturing 50 common workshop tools, such as clamps and screw drivers. Since this dataset presents few objects, but very similar to each other, it can be used to assess the performance of RCFusion in the instance recognition task. For the evaluation, we follow

the standard experimental protocol defined in [7], where training data are collected from fixed viewing angles between the camera and the object while the test data are collected by freely moving the camera around the object.

**OCID:** It includes 96 cluttered scenes representing common objects organized in three subsets: ARID20, ARID10, and YCB10. The ARID20 and ARID10 subsets contain scenes that include, respectively, up to 20 and 10 out of 59 objects from Autonomous Robot Indoor Dataset [9]. Similarly, the YCB10 subset contains scenes with up to 10 objects from Yale-CMU-Berkeley object and model set [31]. Each scene is built incrementally by adding one object at a time and recording new frames at each step. Two ASUS-PRO cameras, positioned at different heights, are used to simultaneously record each scene. Further scene variation is introduced by changing the support plane (floor and table) and the background texture. Since OCID has been acquired to evaluate object segmentation methods in cluttered scenes, semantic labels are not provided by the authors. In order to adapt this dataset to a classification task, we crop out the objects from each frame and annotate them with semantic labels similar to the RGB-D Object Dataset. To avoid redundancies, we go sequentially through the frames of each scene and save only the crops that have an overlap with the bounding box of a newly introduced object. We then filter out the classes with less than 20 images to ensure a minimum amout of training samples per class. We use the crops from the ARID20 subset to train the network for an instance recognition task and then use the crops from the ARID10 subset for testing. Overall, we obtain 3,939 RGB-D images capturing 49 distinct objects. The original datasets, as well as the crops and annotation used in this letter are available at "https://www.acin.tuwien.ac.at/en/vision-for-robotics/software-tools/object-clutter-indoor-dataset/".

### B. Architecture

The network architecture of RCFusion passes through independent design choices of three main elements: RGB-/Depth-CNN, projection blocks and RNN.

**RGB-/Depth-CNN:** With computational and memory efficiency in mind, we choose a CNN architecture with a relatively small number of parameters. Since residual networks have become a standard choice, we deploy ResNet-18, the most compact representation proposed by He *et al.* [32]. ResNet-18 has 18 convolutional layers organized in five residual blocks ($\sim 40,000$ parameters). We extract our features after each skip connection in the network. The network has two skip connections per residual blockand we start extracting from the second block: this results in $L = 8$ extracted features per network. An implementation of ResNet-18 pre-trained on ImageNet is available in [33].

**Projection blocks:** The projection blocks, shown in Figure 3, are designed in such a way that the first convolutional layer focuses on exploiting the spatial dimensions of the input, width and height, with $D = 512$ filters of size $(7 \times 7)$, while the second convolutional layer exploits its depth with $D = 512$ filters of size $(1 \times 1)$. Finally, the global max pooling computes the maximum of each depth slice. This instantiation of the projection blocks has provided the best performances among those that we tried.

TABLE I
ACCURACY (%) OF SEVERAL METHODS FOR OBJECT RECOGNITION ON RGB-D OBJECT DATASET [6]. RED: HIGHEST RESULT; BLUE: OTHER CONSIDERABLE RESULTS

| RGB-D Object Dataset | | | |
|---|---|---|---|
| Method | RGB | Depth | RGB-D |
| LMMMDL [22] | 74.6±2.9 | 75.5.8±2.7 | 86.9±2.6 |
| FusionNet [2] | 84.1±2.7 | 83.8±2.7 | 91.3±1.4 |
| CNN w/ FV [19] | **90.8**±1.6 | 81.8±2.4 | **93.8**±0.9 |
| DepthNet [4] | 88.4±1.8 | 83.8±2.0 | 92.2±1.3 |
| CIMDL [23] | 87.3±1.6 | **84.2**±1.7 | 92.4±1.8 |
| FusionNet enhenced [3] | **89.5**±1.9 | **84.5**±2.9 | **93.5**±1.1 |
| DECO [18] | **89.5**±1.6 | 84.0±2.3 | **93.6**±0.9 |
| **RCFusion** | **89.6**±2.2 | **85.9**±2.7 | **94.4**±1.4 |

**RNN:** In a trade-off between network capacity and small number of parameters, we use the popular GRU [27]. In our experiments, we process the sequence of projected vectors with a single GRU layer with a number of memory neurons $M = 50$. An implementation of GRU can be found in all the most popular deep learning libraries, including tensorflow.

### C. Training

We train our model using RMSprop optimizer with batch size 64, learning rate 0.001, momentum 0.9, weight decay 0.0002 and max norm 4. The architecture specific parameters have been fixed through a grid search to projection depth $D = 512$ and memory neurons $M = 50$. The weights of the two ResNet-18 used as the RGB- and Depth-CNN are initialized with values obtained by pre-training the networks on ImageNet. The rest of the network is initialized with Xavier initialization method in a multi-start fashion, where the network is initialized multiple times and, after one epoch, only the most promising model continues the training. All the parameters of the network, including those defining the RGB- and Depth-CNN, are updated during training. The input to the network is synchronized RGB and depth images pre-processed following the procedure in [3], where the depth information is encoded with surface normals, the best non-learned colorization method (see Section II). For JHUIT-50 and OCID, we compensate for the small training set with simple data augmentation techniques: scaling, horizontal and vertical flip, and 90 degree rotation.

### D. Results

In order to validate our method, we first compare the performance of RCFusion to existing methods on two benchmark datasets, RGB-D Object Dataet and JHUIT-50. We then test our method on a more challenging dataset, OCID, and perform an ablation study to showcase the contribution of each component of the method.

**Benchmark:** We benchmark RCFusion on RGB-D Object Dataset and JHUIT-50 against other methods in the literature. Table I shows the results on RGB-D Object Dataset for the object categorization task. The reported results for the RGB and depth modality are obtained by training a classifier on the final features of the RGB- and Depth-CNN, respectively. The reported

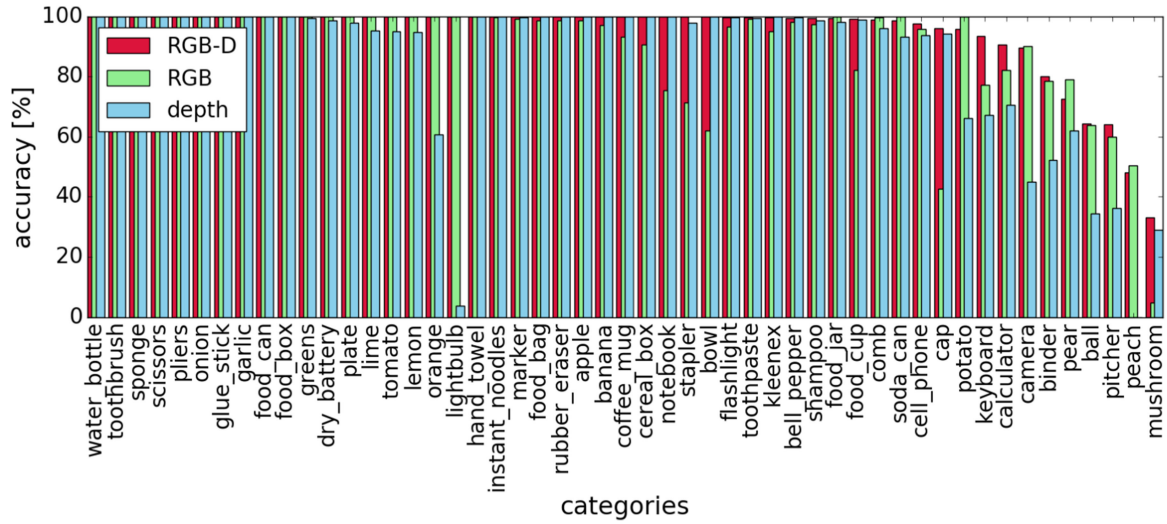


Fig. 4. Per class accuracy (%) of RCFusion on RGB-D Object Dataset [6].

multi-modal RGB-D results show that our method outperforms all the competing approaches. In addition, the results of the single modality predictions demonstrate that ResNet-18 is a valid trade-off between small number of parameters and high accuracy. In fact, on the RGB modality, the accuracy is second only to [19], where they use a VGG network [34] that introduces considerably more parameters than ResNet-18. For the depth modality, ResNet-18 provides higher accuracy than all the competing methods.

In order to gain a better insight on the performance of RCFusion, we consider the accuracy on the individual categories of RGB-D Object Dataset. Figure 4 shows that the multi-modal approach either matches or improves over the results on the single modalities for almost all categories. For categories like "lightbulb", "orange" or "bowl", where the accuracy on one modality is very low, RCFusion learns to rely on the other modality. An interesting insight on the functioning of the method is given by comparing, for each category, which other categories generate the misclassification. Table III indicates, for few example classes, the most frequently misclassified class in the RGB, depth and RGB-D case. When an object class is confused with distinct classes in the individual modalities, like for "keyboard" and "calculator", the RGB-D modality can perform better. However, when an object class is confused with the same classes in both RGB and depth modalities, like for "pear" and "potato", the RGB-D modality can perform slightly worse than the single modalities. This highlights a weakness of the method that will be the subject of future investigations.

Table II shows the results on JHUIT-50 for the instance recognition task. For the individual modalities, ResNet-18 shows again a compelling performance. In the multi-modal RGB-D classification, our method clearly outperforms all the competing approaches with a margin of 2% on the best existing method, DECO [18]. In summary, RCFusion establishes new state-of-the-art results on the two most popular datasets for RGB-D object recognition, demonstrating its robustness against changes in the dataset and the task.

TABLE II
ACCURACY (%) OF SEVERAL METHODS FOR OBJECT RECOGNITION ON JHUIT-50 [7]. RED: HIGHEST RESULT; BLUE: OTHER CONSIDERABLE RESULTS

| JHUIT-50 | | | |
|---|---|---|---|
| Method | RGB | Depth | RGB-D |
| DepthNet [4] | 88.0 | 55.0 | 90.3 |
| FusionNet enhanced [3] | **94.7** | 56.0 | **95.3** |
| DECO [18] | **94.7** | **61.8** | **95.7** |
| **RCFusion** | **95.1** | **59.8** | **97.7** |

TABLE III
MOST FREQUENTLY MISCLASSIFIED CLASSES IN RGB, DEPTH AND RGB-D FOR SELECTED REFERENCE CLASSES

| Misclassification cases | | | |
|---|---|---|---|
| Reference class | RGB | Depth | RGB-D |
| calculator | keyboard | hand towel | hand towel |
| keyboard | calculator | binder | calculator |
| pear | apple | apple | apple |
| potato | lime | lime | lime |

**Challenge:** To evaluate the performance of our method on more robotic-oriented data, we show experiments on OCID. This dataset has been recorded with the specific goal of creating highly cluttered and occluded object scenes (see Figure 6). Since objects are presented in clutter rather than in isolation, using multiple modalities is useful to cope with ambiguous views, thus making OCID particularly relevant to evaluate algorithms for RGB-D object recognition. In addition, its small training set of 2,428 cropped images represents an additional challenge. Table IV shows the results on OCID for the instance recognition task. As well as our method, we also report the results of DECO, that showed competitive performance on RBG-D Object Dataset and JHUIT-50. The results on the single modalities show that the depth data alone are not very informative for this task, with a gap of 50% with respect to the RGB modality. Nevertheless, our method leverages both modalities and obtains an improvement

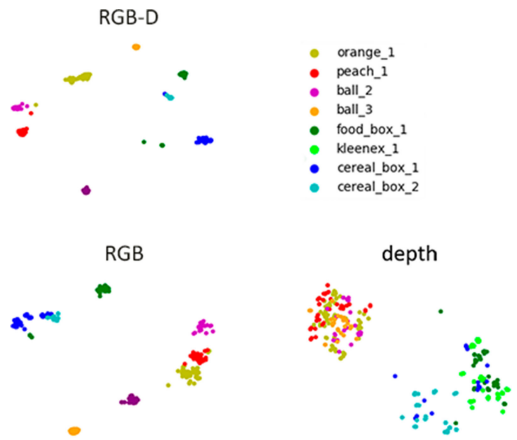


Fig. 5. t-SNE visualization of the final features obtained for RGB, depth and RGB-D modalities.


Fig. 6. Examples of object crops from the Object Cluttered Indoor Dataset [8] with their instance label.

of 6.1% in accuracy with respect to the RGB modality alone. On the contrary, DECO reveals its limits and maintains the same performance of the RGB modality even in the multi-modal case. This result is due to the simple strategy used in DECO for the multi-modal fusion: the final prediction is made by selecting the class with the maximum probability among the RGB and depth predictions. The more complex modality fusion of RCFusion thus translates into a non-trivial improvement of over 10% in accuracy with respect to DECO.

**Feature analysis:** An interesting intuition of the effectiveness of RCFusion comes from the visualization of the features learned on the OCID dataset. Figure 5 represents the two dimensional t-SNE embedding of the final features of the different modalities. As expected, the t-SNE embedding of the depth features clusters together objects with similar shapes. For example, objects with near-spherical shapes like "orange_1", "pear_1" and "ball_2(/3)" are grouped together. The RGB modality provides more discriminative features, but similar pairs of objects, like ("orange_1"-"peach_1") and ("cereal_box_1"-"cereal_box_2") are very close to each other. Instead, the embedding of the RGB-D features neatly separates each object in discernible clusters.

**Ablation study:** To observe the contribution of the two main elements of RCFusion, multi-level feature extraction and recurrent fusion, we alternatively remove these elements and compare the performance with the full version of the method. Table IV presents the results of these variations on OCID. It can be noticed that using only the features from the last layer of the RGB-/Depth-CNN (RCFusion - res5) drops the performance by 2% in accuracy. This confirms that explicitly using features from several levels of abstraction improves the multi-modal recognition compared to only using the final features of single modalities. Analogously, if instead of using the RNN we concatenate the multi-modal features from the projection blocks and fuse them with a fully connected layer, the performance drops by 3.1% in accuracy. This confirms that a more sophisticated fusion mechanism that effectively combines the modalities while retaining the crucial information from the different levels of abstraction is crucial for obtaining a final discriminative RGB-D feature.

TABLE IV
ACCURACY (%) OF DECO [18] AND VARIATIONS OF RCFUSION ON OBJECT CLUTTER INDOOR DATASET [8]. "RCFUSION - RES5" IS THE VARIATION OF RCFUSION WHEN ONLY THE FEATURES FROM THE LAST RESIDUAL LAYER (RES5) ARE USED FOR CLASSIFICATION. "RCFUSION - FC" IS THE VARIATON OF RCFUSION WITH A FULLY CONNECTED LAYER USED INSTEAD OF THE RECURRENT NEURAL NETWORK FOR COMBINING THE RGB AND DEPTH FEATURES

| Object Clutter Indoor Dataset | | | |
|---|---|---|---|
| Method | RGB | Depth | RGB-D |
| DECO [18] | **80.7** | **36.8** | 80.7 |
| RCFusion | **85.5** | **35.0** | **91.6** |
| RCFusion - res5 | - | - | **89.6** |
| RCFusion - fc | - | - | 88.5 |

## V. DISCUSSION AND CONCLUSION

In this letter, we have presented RCFusion: a multi-modal deep neural network for RGB-D object recognition. Our method uses two streams of convolutional networks to extract RGB and depth features from multiple levels of abstraction. These features are concatenated and sequentially fed to an RNN to obtain a compact RGB-D feature that is used by a softmax classifier for the final classification. We show the validity of our approach by outperforming the existing methods for RGB-D recognition on two standard benchmarks, RGB-D Object Dataset and JHUIT-50. We also stress test RCFusion with some of the main challenges of robotic vision by evaluating it on OCID. In fact, not only does this dataset present highly cluttered and occluded scenes, but it also provides few training samples. Despite these challenges, RCFusion presents compelling results on OCID and marks the superiority of our multi-modal fusion mechanism.

Our experiments also show that there is space for improvement. For the classes where the predictions of the RGB and the depth modality are often fooled by the same class, RCFusion can perform worse than the single modalities. In future work, we will investigate how to mitigate this problem by using an ensemble of different fusion mechanisms.

The novel architectural choices of this letter are presented in their simplest instantiation to promote the validation of the

underlying idea rather than a specific implementation. In future work, we will evaluate more complex configurations of recurrent networks for the multi-level sequential fusion. Due to their implementation-agnostic nature, the main concepts presented in this letter can be adapted to different tasks. The results obtained on object categorization encourage further research to extend this approach to higher level tasks, such as object detection and semantic segmentation.

## REFERENCES


[1] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[2] A. Eitel, T. Springenberg, L. S. M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 681–687.

[3] A. Aakerberg, K. Nasrollahi, and T. Heder, "Improving a deep learning based RGB-D object recognition model by ensemble learning," in *Proc. IEEE 7th Int. Conf. Image Process. Theory, Tools Appl.*, 2017, pp. 1–6.

[4] F. Carlucci, P. Russo, and B. Caputo, "A deep representation for depth images from synthetic data," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1362–1369.

[5] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS Workshop Deep Learn.*, 2014.

[6] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1817–1824.

[7] C. Li, J. Bohren, E. Carlson, and G. D. Hager, "Hierarchical semantic parsing for object pose estimation in densely cluttered scenes," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 5068–5075.

[8] M. Suchi, T. Patten, and M. Vincze, "EasyLabel: A semi-automatic pixel-wise object annotation tool for creating robotic RGB-D datasets," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019.

[9] M. Loghmani, B. Caputo, and M. Vincze, "Recognizing objects in-the-wild: Where do we stand?" in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 2170–2177.

[10] Tensorflow, Accessed: Feb. 4, 2018. [Online]. Available: https://www.tensorflow.org/

[11] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 821–826.

[12] L. Bo, X. Ren, and D. Fox, "Hierarchical matching pursuit for image classification: Architecture and fast algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2115–2123.

[13] M. Blum, J. T. Springenberg, J. Wülfing, and M. Riedmiller, "A learned feature descriptor for object recognition in RGB-D data," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 1298–1303.

[14] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3D object classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 656–664.

[15] J. Deng, W. D. R. Socher, L. Li, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[16] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proc. Comput. Vision—Eur. Conf. Comput. Vis.*, 2014, pp. 345–360.

[17] L. Bo, X. Ren, and D. Fox, *Unsupervised Feature Learning for RGB-D Based Object Recognition*. Berlin, Germany: Springer, 2013, pp. 387–402.

[18] F. M. Carlucci, P. Russo, and B. Caputo, "$(DE)^2$ CO: Deep depth colorization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2386–2396, Jul. 2018.

[19] W. Li, Z. Cao, Y. Xiao, and Z. Fang, "Hybrid RGB-D object recognition using convolutional neural network and fisher vector," in *Proc. Chin. Autom. Congr.*, 2015, pp. 506–511.

[20] S. Tang *et al.*, "Histogram of oriented normal vectors for object recognition with a depth sensor," in *Proc. Asian Conf. Comput. Vis.*, 2013, pp. 525–538.

[21] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.

[22] A. Wang, J. Lu, J. Cai, T. J. Cham, and G. Wang, "Large-margin multi-modal deep learning for RGB-D object recognition," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1887–1898, Nov. 2015.

[23] Z. Wang, R. Lin, J. Lu, J. Feng, and J. Zhou, "Correlated and individual multi-modal deep learning for RGB-D object recognition," 2016, arXiv:1604.01655.

[24] Hariharan *et al.*, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 447–456.

[25] Bell *et al.*, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. IEEE Conf. Comput Vis. Pattern Recognit*, 2016, pp. 2874–2883.

[26] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[27] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] J. Schmidhuber and S. Heil, "Sequential neural text compression," *IEEE Trans. Neural Netw.*, vol. 7, no. 1, pp. 142–146, Jan. 1996.

[30] M. Mahoney, "Fast text compression with neural networks," in *Proc. 13th Int. Florida Artif. Intell. Res. Soc. Conf.*, 2000, pp. 230–234.

[31] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the yale-CMU-berkeley object and model set," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 36–52, Sep. 2015.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[33] Pretrained ResNet-18, Accessed: Feb. 21, 2018. [Online]. Available: https://github.com/HolmesShuan/ResNet-18-Caffemodel-on-ImageNet.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Repres.*, 2015.