Learning joint features for color and depth images with Convolutional Neural Networks for object classification

Eder Santana, Karl Dockendorf, Jose C. Principe

Abstract— **In this paper we investigate the advantages of learning representations of color plus depth images (Red-Blue-Green-Depth, RGB-D) over color only images (RGB) for computer vision. Specifically, we investigate the advantages on the task of object recognition. For this purpose, we applied the state-of-art deep convolutional neural networks (CNN) for classification of images on the RGB-D dataset published by [1]. We show that this approach provides better results than those that use separate features for color and depth. Also, we probe the resulting CNN to gain intuition about how filters for depth and color channels iterate to generate useful features.**

I. INTRODUCTION

Object recognition is one of the main areas of study in computer vision. In order to automatically provide labels for pictures of objects, several approaches have been taken throughout the years. The approaches that attracted most attention were the ones based on scale and shift invariant features extractors, for example SIFT [2], SURF [3], HOG [4] and ORB [5].

In recent benchmarks, those features extraction methods were surpassed by deep neural networks that learn the features and the classifiers straight from data [6]. The first approaches to train to deep neural networks, were based on pre-training the individual layers through unsupervised learning and then fine-tuning the final model using supervised back propagation. Notably, the most successful methods for the unsupervised step were based on Restricted Boltzman Machines [7], Auto-Encoders [8] and Dynamical Systems [9]. Recently, Andrew Ng and colleagues showed thorough several papers that methods like ICA, Sparse Filtering and even simple k-means can be successfully used for unsupervised pre-training [10].

Nevertheless, one of the most astonishing results of neural networks for the computer vision community were obtained during the Imagnet competition at ILSCVRC 2012 through plain supervised backpropagation. Largely due to the availability of enough data (1.5 Terabytes of images) and powerful regularization techniques such as Dropout [6], Krizchevizk et. al. [11] were able to train a neural network classifier composed of 7 layers that obtained an accuracy of $\sim$86% and beat the benchmark on the Imagenet dataset. The second position was a submission based on a combination of SIFT and other descriptors that obtained a lower accuracy of $\sim$74%. Also, CNNs like those trained on the Imagenet dataset can provide better features than those of the other aforementioned methods. Razavian et. al. [12] showed that CNN-based features provide state-of-art results on several other computer vision benchmarks as well.

Nowadays, the necessity of unsupervised pre-training has become controversial due to the difficulty of defining a proper unsupervised technique, determining how much pre-training is necessary, and the success of full back propagation. Practical issues also motivate the necessity of training CNNs in supervised mode only, such as the minimal gain in accuracy [6] and large datasets, which would consume too much time in an unsupervised training phase. Based on those arguments, here we also opt in for training CNNs using only back propagation.

Meanwhile, the Microsoft Kinect project made depth and 3D vision sensor ubiquitous thus garnering attention toward developing systems that exploit this extra dimension of image data. This, combined with the recent advances in computer vision that were made possible due to the availability of large datasets, [1] published a large collection of color plus depth images, referred as UW RGB-D dataset (RGB for color and D for depth channels).

Unfortunately, all the methods that analyzed the UW RGB-D dataset for the task of object recognition applied feature extractors, either hand engineered [1] or learned from data [13], separately to color and depth channels. For instance in [1] the authors proposed a combination of HOG and SIFT features for modeling the color image and "spin-images" [14] features for modeling the depth images. Another approach proposed learning features using k-means and sparse coding [13], again, individual models were trained for each data sensor. Even more innovative architectures were proposed in [15], where recursive convolutional layers projects the data to high dimensional spaces in an approach similar to Extreme Learning Machines. As one can see, this implies modeling the color and depths channels as independent and the joint distribution as the product of marginals. Even intuitively, one can see that this model does not hold true because the colors of each pixel in the image is not statistically independent of the distance of those pixels to the camera, as it is well known in color and aerial perspective theory [16]. Another argument against the mentioned method is that they can be seen as deep models without the supervised fine-tuning of the first layers, only the top most logistic regression layer is adapted by backpropagation.

Contrary to the previous works in the literature, here we propose learning features that jointly represents RGB-D images without the implicit independence assumption. Thus, here we model each image as a tensor with the dimensions

Research realized during Summer, 2014 at Paracosm (http://paracosm.io).

defined by the images width, height and 4 channels (3 for color and 1 for depth). Those images and their respective labels are used as input for training a CNN. This way, the CNN learns distributed features for both color and depth. Those features should be readily used through all the nonlinear steps implemented by the layers of the deep CNN for better recognition which is what backpropagation theoretically guarantees [17]. In the next sections, we show that this approach provides better classification results at the instance classification task on the UW RGB-D dataset.

The following sections of this paper are organized as follow. Section II details the dataset and the object recognition problem at hand. We devote Section III to describing the CNN architecture the we applied for learning features from RGB-D images. In Section IV, we compare the results that we obtained with those previously published for the same problem as well as similar architectures learning from RGB only images, we also probe the neural network filters and preferred stimulus at this section with the purpose of gaining intuition about what the CNN considered helpful for recognition. We conclude this paper in Section V.

II. Object recognition from RGB-D images

The RGB-D dataset [1] consists of 300 everyday use objects such as coffee mugs, computer keyboards, soda cans, and tomatoes. Several pictures of those objects were taken from different angles, rotations and distances using a PrimeSense camera. That camera consists of a technology similar to the one present on the famous Microsoft Kinect sensors. Those pictures consists of a color (RGB) and depth. The depth information is calculated from infrared sensor measures as scaled units of distance, but the detailed algorithm used by the device is undisclosed. At a high level description, we can say that the infrared grid of dots is projected and the deformation of the grid viewed by the paired camera is used to interpret the depth. Samples of pictures from that dataset, along with classification results are show in Fig. 1. Here, we investigated the task of classifying independent frames in one of 300 classes. It is noteworthy that the previously reported approaches used only a small portion of the training data for unsupervised pre-training. This was due to the fact that unsupervised learning usually overfits the most repetitive aspects of the training data (such as edges and other high contrast patches in this case) [13]. On the other hand, here we used Dropout [6] to avoid overfitting, which allowed to use the full training dataset (part of which was kept for validation, the test dataset was never seen during training).

III. Convolutional Neural Network architecture

The CNN architecture that we used is similar to the one proposed for classifying the CIFAR-10 dataset [6]. The CIFAR-10 consists of 70000 color images of 32 x 32 pixels, divided in 10 classes. Still inspired by that approach, we reshaped all the images at the UW RGB-D dataset to 32x32 and removed the mean, which also sped up the training time by a few hours.

The CNN itself consists of the following layers and processing steps:

1) Convolutional layer, 64 filters of size 5x5 each,
2) Max-pooling, 3x3 pool, stride of 2 and input is padded with 2 rows and 2 columns of zeros
3) Cross-channel normalization,
4) Convolutional layer, 64 filters of size 5x5
5) Max-pooling, 3x3 pool and stride of 2,
6) Cross-channel normalization
7) Fully-connected layer (regular MLP layer), 2056 outputs,
8) Fully-connected layer, with 2056 outputs, and
9) Fully connected layer, that outputs 300 classes

It is important to comment that convolutional layers are implemented as following. Let $\mathbf{X}_c$ be an input image with composed the channels $c = 1, 2, 3, 4$ that addresses red, blue, green and depth, respectively. A convolutional filter W_c with the same number of input channels as $\mathbf{X}$ produces an output defined by:

$$\mathbf{O} = \sum_{c=1}^{4} \mathbf{X}_c \star \mathbf{W}_c, \tag{1}$$

where $\star$ denotes the convolution operation.

An attentive reader may notice that if the scale of the four channels are different, the summation in 1 may contain biased information to only one of those channels. He may even be concerned that this effect is propagated forward through all the convolutional layers. For instance, in some cases the color channels are encoded as 8 bits unsigned integers (maximum value 256) and depth channels as 16 bits unsigned integers (maximum value 65535). This is not the case for the RGB-D dataset where both color and depth channels are equally encoded as uint8. Nevertheless, we further refer to the Cross-channel normalization (also referred as cross-map normalization, or cmnorm for short) steps referred at items 3 and 6 above. This step consists of the following nonlinearity.

Let $\mathbf{O}_i$ be the i-th output channel of a convolutional layer. Calculate the $\mathbf{R}_i = (1+\alpha \sum_{j=-k}^{k} (O_{i+k})^2)^\beta$ and output $\frac{\mathbf{O}_i}{\mathbf{R}_i}$. Where k, α, β are free parameters defined by the user. Here we kept $k = 3$, $\alpha = 0.001$ and $\beta = 0.75$ as suggested by [6]. But note that this normalization enhances the competition between filters and distributes the representations, in order to guarantee that different filters do not overfit or over explain the same input patterns. Also, through backpropagation of the errors, the stochastic gradient optimization is capable of scaling the filters $\mathbf{W}_c$ in order to account for a proper scaling of the input channels without the user necessarily preprocessing the data. This intuition is confirmed by our results shown in the next section.

Further, we applied a rectified linear unity as the activation function after the convolutional and fully connected layers. The final classification result of the network is calculated as

the maximum of a softmax nonlinearity applied over the last layer output vector $\mathbf{z}$:

$$y_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{2}$$

The cost function for training the network is the negative log-likelihood between the estimated labels y and the desired labels t:

$$L(y,t) = \sum_{k=1}^{K} -t_k log(y_k) \tag{3}$$

We simulated the network on a NVIDIA GPU GTX 650 using cuda-convnet library [1]. We added a data handler to the library in order to deal with 4 channel (RGB-D) input images, we also compared the results using color only input and the data handler provided by the CIFAR-10 experiment. We report our results at the next section.

IV. Results

We run the network mentioned on the previous section for 200 epochs and kept the best solution by cross-validation. We separated every fifth frame as in the protocol suggested by [1]. In Table I we show our results alongside the results published before on the same dataset. The architecture proposed in this paper is referred as CNN+RGB-D for color plus depth images and CNN+RGB for color only input images.

TABLE I
Classification accuracy (% correct) on UW RGB-D dataset instance recognition task.

CNN+RGB-D	CNN+RGB	[1]	[13]
99%	96%	90.2%	92.8%

Some classification results using color and depth input images are show in Fig. 1.

Beyond that, to confirm our intuition that training CNNs on RGB and depth channels together leads to joint feature representations, we show some of the preferred stimulus that leads to the classification of 3 different classes in Fig. 2. There, we can see that the trained CNN looks for patterns at the 4 input channels at the same time. For the class apple-1 we can see that the preferred stimulus is relatively smooth in the center of the image, as compared to the keyboard-1 preferred stimulus. The later presents high frequency structures, possibly related to the keys of the keyboard. Also, in all the preferred stimuli, we see information about the preferred backgrounds. This is due to the fact that the dataset does not have diversity enough. Thus, we believe that this is used by the network to calculate the relative size of the objects in the scene. Note how the network expects that food-can-12 and apple-1 have depth over the average in the upper corners (white values).

To confirm that the network learns to deal by itself with the scales of the input channels, we investigated the

[1]https://code.google.com/p/cuda-convnet/

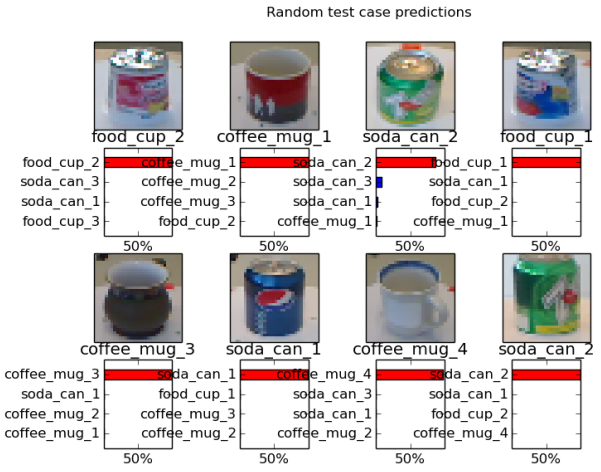

Fig. 1. Sample of classification results of the CNN dataset using RGB-D features. Each subplot shows the first five guesses of the algorithm. The red bar indicates the right label.

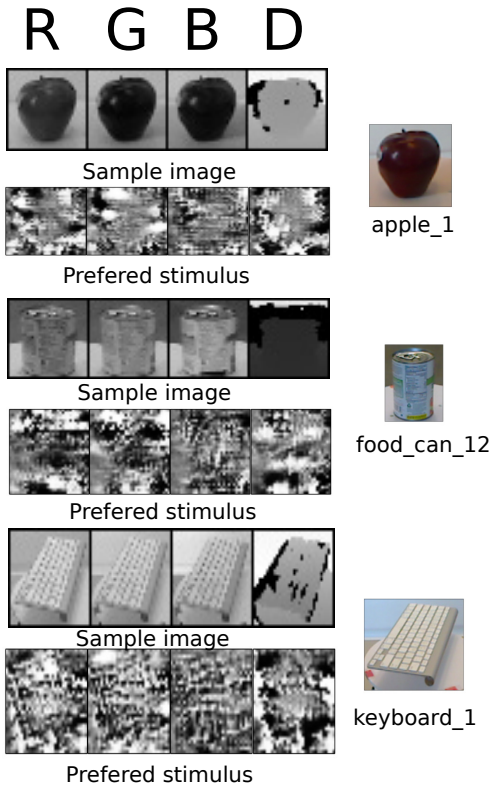

Fig. 2. Preferred input image of our RGB-D CNN for 3 different classes in the UW RGB-D dataset. Note that those stimulus are in the zero mean space that is the actual input of the network. Thus, their values should be interpreted relatively to the mean. White represents values over the mean and black, values under the mean.

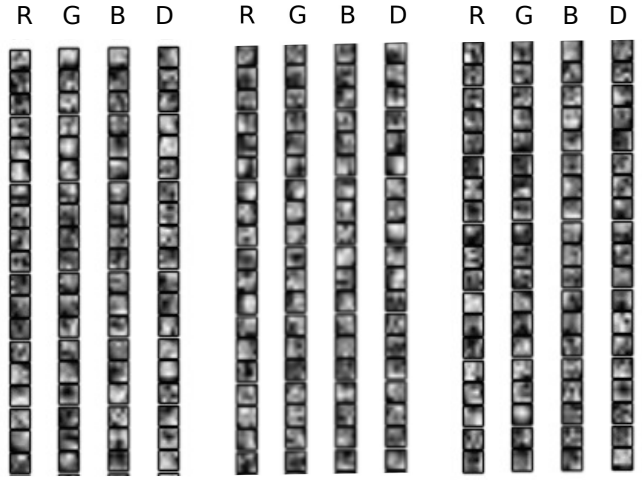

Fig. 3. Convolutional kernels of the first layer of our RGB-D CNN. Note the that some filters learns Gabor like edges that span through all the four channels.

filters of the first layer of the network. We show the 64 learned filters in Fig. 3. Also, we investigate the norm of those filters. Particularly, we noticed that some filters have all the 4 channels in the same scales, while others have the depth channel 10 times smaller than the color channel (the filter in the fifth row of Fig. 3). The distribution of norms per channel This confirms our intuition that plain backpropagation is capable of dealing the scales of the input images, not requiring preprocessing on that respect.

V. Conclusions

In this paper we investigated the advantages for learning joint features for color and depth images for object recognition. We showed that better results can be obtained when a large convolutional neural network is trained on the full images, where the depth channel is treated on equal bases as the color channels. Thus, the network was able to learn features, under supervision, that do not rely on the assumption of independence between RGB and depth images.

We obtained an accuracy of 99% using RGB-D while we got only 96% accuracy using RGB only, although we used the same CNN architecture for both cases. Also, we note that both results are higher than those previously published on the same task, which restates the superiority of deep learning based methods for object recognition.

Future applications of the present work may include a system for online object recognition on a market shelf. We note that recent advances in the literature points to the possibility of the same CNN used for recognition to be used for segmentation [18]. We plan to investigate the advantages of the joint color and depth features learned in the present research in the segmentation problem.

References

[1] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 1817–1824.

[2] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 1999, pp. 1150–1157 vol.2.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2007.09.014

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, Jun. 2005, pp. 886–893 vol. 1.

[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov. 2011, pp. 2564–2571.

[6] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[7] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[8] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1096–1103.

[9] J. Principe and R. Chalasani, "Cognitive architectures for sensory processing," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 514–525, April 2014.

[10] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 215–223.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., 2012, pp. 1106–1114. [Online]. Available: http://books.nips.cc/papers/files/nips25/NIPS2012_0534.pdf

[12] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN Features off-the-shelf: an Astounding Baseline for Recognition," *CoRR*, vol. abs/1403.6382, 2014.

[13] 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.

[14] A. E. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999. [Online]. Available: http://dx.doi.org/10.1109/34.765655

[15] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng, "Convolutional-Recursive Deep Learning for 3D Object Classification."

[16] L. da Vinci and J. P. Richter, *The Notebooks of Leonardo da Vinci*. Dover Publications, 1970, vol. 1.

[17] J. C. Principe, N. R. Euliano, and W. C. Lefebvre, *Neural and Adaptive Systems: Fundamentals Through Simulations with CD-ROM*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1999.

[18] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, August 2013.