UPDATE: Object Detection Enhancement with RBG-D Data

Megan Egbert, Arman Karimian, Noushin Mehdipour, Tayler Pauls, Athar Roshandelpoor
{megbert,armandok,noushinm,tayler,athar}@bu.edu

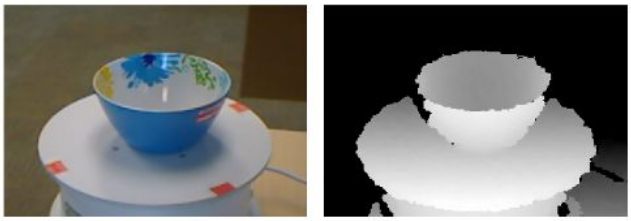

Figure 1. RGB and depth images. The RGB image provides appearance and texture of the object (left). The depth image provides shape and metric information, irrespective of lighting conditions (right). *Figure adapted from RGB-D Object Dataset* [2].

1. Task

Many image classification algorithms are based entirely on red-blue-green (RGB) images. These RGB images provide the appearance and texture of the object(s) to be classified. While RGB image classifiers present impressive results, adding depth information, in the form of RGB-D images, to the classifier yields even better results because it gives robust information about the shape (and size) of the object(s), see Figure 1 [1]. The pixel values in a depth image represent the metric distance of actual points in front of a camera, from which the approximate size of an object can be extracted. In this project, we will introduce a light-weight convolutional neural net with two parallelly trained models, one for RGB data and one for depth images, and fuse them together for a comprehensive RGB-D object detector. For the RGB part, we will make use of a pre-trained mobilenet model. For the depth part, we will introduce a novel structure and train it from scratch. Creating the depth model is particularly challenging due to the lack of labeled depth images, thus we will have to initialize the model with RGB parameters, as done by Eitel et. al. [1]. Lastly, we will converge the two models via fully connected layers at the end of the deep learning network. With this RGB-D object detector, assorted with a Single Shot multi-box Detector or R-CNN, we will demonstrate the classification results improve relative to the isolated RGB model.

2. Related Work

The concept of expanding RGB networks to include depth information in the form of RGB-D images has been evaluated by many groups. Eitel and colleagues create a two-stream convolutional neural network, where one stream is an RGB-trained model, and the other is a depth-trained model initialized with RGB model parameters [1]. Eitel also adds artificial noise to the depth images to enhance performance [1]. Alternately, Gupta and colleagues use hand-crafted features to represent RGB-D images to compensate for the limited data, which is insufficient for *ab initio* CNN training [3]. Specifically, they use the HHA encoding for depth data which includes horizontal disparity, height above ground, and angle with the direction of gravity [3]. These features are used to adapt an RGB pre-trained model to reflect RGB-D data. Gupta and colleagues show that this geocentric embedding for depth images in a Region-CNN (R-CNN) outperforms deep learning from raw RGB-D images [3]. However, Song and colleagues argue that there are important differences between RGB and HHA data (i.e. Gabor-like patterns and high-frequency patterns), and thus argue against using an RGB pre-trained model for depth images [4]. Instead, Song and colleagues focus on fine-tuning the bottom layers of the network, overcoming the vanishing gradient problem which arises when tuning an RGB model with HHA data [4].

Overall, these methods all start from an RGB pre-trained model and use depth information to fine-tune the model. The differences lie in the availability of the depth data and the fine-tuning methodology. In this project, we will work on a model similar to one proposed in [1], but replace the RGB object detector with a pre-trained mobilenet model due to its efficiency, and fuse its final layers with the output of a novel CNN which takes the depth image as input. Our goal is to make use of the metric information in the depth image to be able to distinguish the size of the objects. This will be useful in comparing similar objects with different size, i.e. a car and a toy car.

3. Approach

Our network consists of two separate networks, one for RGB image and one for depth image, both of which are joined together in the final layers (see Figure 2). RGB stream has a deep CNN model that has been pre-trained for object classification task on the ImageNet dataset before. Using pre-trained RGB networks is common in RGB-D object detection due to the limited available training datasets with depth information.

In our first approach, we finetune the parameters of the RGB network on our target dataset while training the Depth network and the final joining layers simultaneously. We use raw depth images as inputs in this approach. It is known that this method yields mediocre results.

To fully leverage the power of CNNs pre-trained on ImageNet we need to preprocess the RGB and depth input data to be compatible with the chosen pre-trained RGB network.

The first preprocessing step is scaling the images to appropriate size. But, this process is detrimental to object recognition performance (loss of shape information), so we need to use a different preprocessing approach. After the scaling step, RGB images can be used in a way they are, but for the depth images, we need additional steps. To make it clear, a network trained on ImageNet has been trained to recognize objects that follow a specific input distribution, which is not the same thing as data coming from a depth sensor. Even so, a lot of features that qualitatively appear in RGB images, such as edge, corner, shaded region, are also visible in the depth images. This realization has led to the idea of using a rendered version of the depth data as input for CNNs trained on ImageNet.

For our second approach, we are considering a preprocessing step on the depth images. An effective and computationally inexpensive method for encoding of depth to color images was proposed in [1]. This colorization method, applied on depth images, is as follows:

We normalize all depth values to lie between 0 and 225. Then, we apply a jet colormap on the given image that transforms the input from a single to a three channel image(colorizing the depth). For each pixel (i,j) in the depth image d of size W*H, we map the distance to the color values ranging like red-green-blue for near to far. Which is essentially showing depth distribution over all three RGB channels. By using this method, Edges often correspond to objects boundaries. The colorization procedure has enough common structure as an RGB image to learn suitable feature representation from the network.

In our third approach, we train the whole network from scratch so as to compare the results with the two previous approaches, where we used a pre-trained image for the RGB network.

We are considering two different network architecture. The first one is inspired by the network presented in [1], which is shown in Figure 2.

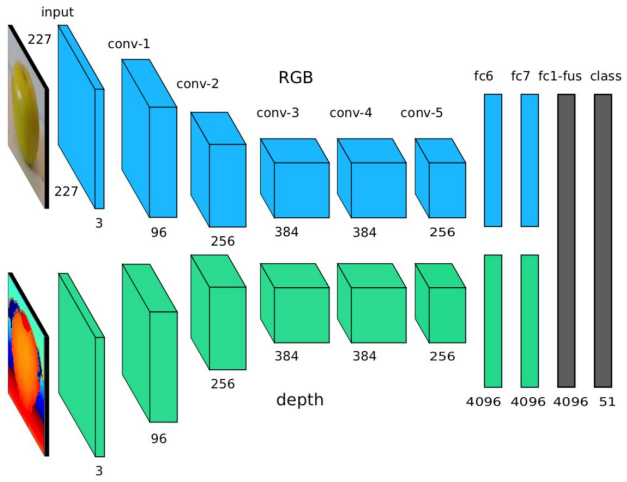

Figure 2. Multimodal /First network architecture

This network works well but it is not very memory efficient. So for our second architecture, we are considering to improve the network architecture and use a variant of MobileNet structure, which is presented in [5].

The MobileNet model is based on depth-wise separable convolutions, a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1 × 1 convolution called a pointwise convolution [5]. The MobileNet structure is built on depth-wise separable convolutions except for the first layer which is a full convolution. The MobileNet architecture is shown in Figure 3. All layers are followed by a batchnorm and ReLU nonlinearity with the exception of the final fully connected layer which has no nonlinearity and feeds into a softmax layer for classification.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figure 3. MobileNet/Second Network Body Architecture

We are working on implementing these Networks for depth images, but we may need to simplify it. Also, we are trying to find a better/new network structure for the depth stream.

4. Dataset and Metric

We evaluate our multimodal network architecture on the Washington RGB-D Object Dataset which consists of 300 household objects (aka "instances") belonging to 51 different categories. As an additional experiment – to evaluate the robustness of our approach for classification in real-world environments – we considered the classification of objects from the RGB-D Scenes dataset whose class distribution partially overlaps with the RGB-D Object Dataset. The RGB-D Scenes Dataset consists of 14 scenes containing furniture (chair, coffee table, sofa, table) and a subset of the objects in the RGB-D Object Dataset (bowls, caps, cereal boxes, coffee mugs, and soda cans). The proposed data augmentation technique for depth data can be used for robust training. We augment the available training examples by corrupting the depth data with missing data patterns sampled from real-world environments. This improves object recognition accuracy in a challenging real-world and noisy environment and shows a better robustness with respect to the RGB-D Scenes dataset. To fully leverage the power of CNNs pre-trained on ImageNet, we pre-process the RGB and depth input data such that it is compatible with the kind of original ImageNet input. We are also considering other possible datasets for train/test the approach.

RGB-D object dataset does not contain accurate camera poses, while “BigBIRD” [6], altough smaller in size, is captured with calibrated Kinect and has 600 images, sampling the viewing hemisphere. “KIT object database” [7] has 196188 images of 145 objects. In “A dataset of Kinect-based 3D scans” [8], a total of 59 small-sized objects (toys) were scanned sequentially from multiple viewing angles using a turntable.

“A large data set of object scans” [9] is the largest dataset of real-life objects by two orders of magnitude which has 10,000 items ranging in size from books to cars. These datasets offer multiple unoccluded views of the same object from different angles.

Dataset	#Objects
RGBD Object Dataset	300
KIT object database	145
A dataset of Kinect-based 3D scans	59
BigBIRD dataset	125
A large dataset of object scans	>10,000

we hope to show that the new RGB-DEPTH combined approach can improve classification accuracy on the test set compared to the previous works.

5. Preliminary Results

So far, we were able to use a pre-trained mobilenet model object classifier and object detector (with SSD). We are in the middle of fine-tuning this pre-trained the initial model on the Washington dataset.

In Figure 4, the output of a MobileNet is shown on an image of a panda.

In Figures 5 and 6, another MobileNet output is shown on the Single Shot Detector region proposals.

Figure 4. Results of a tested image on mobilenet_V2_1.0_224 :
Top 1 prediction: 389 giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca
Probability: 0.85

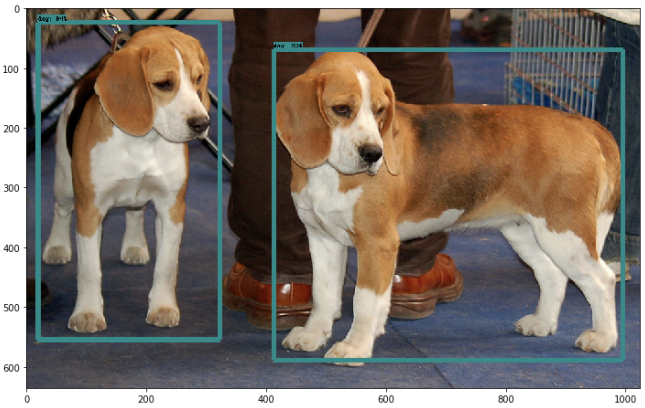

Figure 5. Result of a tested image on mobilenet_V1_coco_2017_11_17 using a Single Shot multibox Detector

Figure 6. Result of a tested image on mobilenet_V1_coco_2017_11_17 using a Single Shot multibox Detector

6. Detailed Timeline and Roles

Task	Deadline	Lead
Implement proposed model	11/25/17	Megan/Athar
Propose a new structure for depth	11/22/17	Arman
Preprocess datasets	11/25/17	Noushin
Prepare plots and results	12/05/17	Tayler
Prepare report and presentation	12/10/17	all

References

1) Eitel, A., Springenberg, J. T., Spinello, L., Riedmiller, M., & Burgard, W. (2015, September). Multimodal deep learning for robust rgb-d object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*(pp. 681-687). IEEE.
2) Lai, K., Bo, L., Ren, X., & Fox, D. (2011, May). A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*(pp. 1817-1824). IEEE.
3) Gupta, S., Girshick, R., Arbeláez, P., & Malik, J. (2014, September). Learning rich features from RGB-D images for object detection and segmentation. In *European Conference on Computer Vision* (pp. 345-360). Springer, Cham.
4) Song, X., Herranz, L., & Jiang, S. (2017, February). Depth CNNs for RGB-D Scene Recognition: Learning from Scratch Better than Transferring from RGB-CNNs. In AAAI (pp. 4271-4277).
5) G. Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.
6) A. Singh, J. Sha, K. Narayan, T. Achim, and P. Abbeel. BigBIRD: A large-scale 3D database of object instances. In *International Conference on Robotics and Automation (ICRA)*, 2014
7) A. Kasper, Z. Xue, and R. Dillmann. The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. *International Journal of Robotics Research*, 2012.
8) A. Doumanoglou, S. Asteriadis, D. S. Alexiadis, D. Zarpalas, and P. Daras. A dataset of Kinect-based 3D scans. In *3D Image/Video Technologies and Applications*, 2013.
9) S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016.