CS 543 - Progress Report

Dario Aranguiz
Cu-Khoi-Nguyen Mac

April 9, 2016

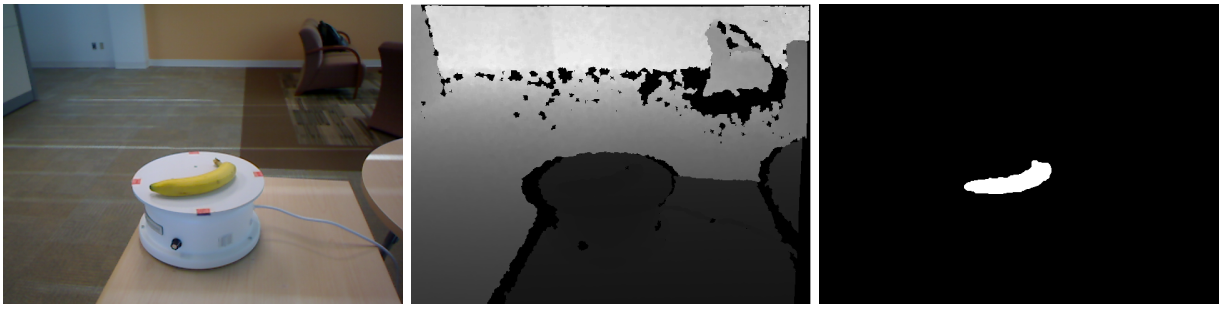
1 Updated statement of the project definition and goals

2 Current member roles and collaboration strategies

3 Proposed approach

4 Data

Following the proposed approach by Eitel et al. [1], we decide to use Washington's RGB-D object dataset [2] for training and testing. The dataset contains 300 household objects (instances) divided into 51 categories. Each instance has three different takes, each records a full turn-around rotation of the object. A sample of the full database includes images of color, depth, mask (as seen in Figure 1), and a text file recording the top-left corner of the object.



(a) Color image (b) Depth image (c) Mask

Figure 1: A banana sample of Washington's RGB-D object dataset. The depth image is rescaled to the range [0, 255] to visual the information.

To speed up the process of testing our system's runnability, we pick out 15 instances from 5 different categories (3 each), namely `apple`, `ball`, `banana`, `bell_pepper`, and `binder`. After the system is fully functional, we will expand the training list to the whole dataset.

5 Initial results

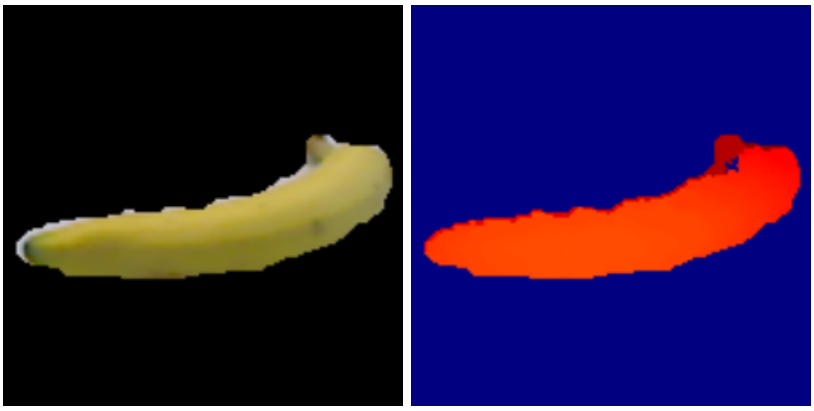
5.1 Data preprocessing

We use the provided masks and top-left corners to remove unnecessary information from color and depth images (Figure 2). Top bottom-right corner if dynamically found, depending on each sample. According to Eitel et al. [1], they convert the depth images into color ones using colorjet mapping as they want to have the same training mechanism for both color and depth. We inherit this idea and apply the same depth

colorizing technique. The images are then rescaled to 227x227 (Figure 3) by replicating the longer side from the cropped images. This help to maintain the object in the center without deformation after rescaling.

(a) Color image (b) Depth image

Figure 2: Cropping banana sample after applying mask.



(a) Color image (b) Depth image (after colorizing)

Figure 3: Rescaling banana sample to 227x227 by replicating the longer side.

5.2 Model architecture

Figure 4 illustrates the network architecture proposed by Eitel et al. [1]. There are two separate stream models, corresponding to color and depth channels. Each stream contains 5 convolution (from `conv-1` to `conv-5`) and 2 fully connected layers (`fc6` and `fc7`). The streams are fused together by one fusion layer (`fc-fus`) and classified by `class` layer.

Figure 5 shows our model, constructed by following the description in Figure 4. The model is plotted using visualizer package of Keras. Using Keras, each stream is constructed using `Sequential()` model and the fusion layer is created by `Merge()` layer. We propose to augment each convolutional layer (`Convolution2D()`) with a maxpooling (`MaxPolling2D()`) and a dense sampling layer (`Dense()`) to reduce the risk of over-fitting and decrease memory usage.

In fact, there are two different training process: one to find the weights of stream model and one for the fusion model. To train a stream model, we create an architecture similar to a branch in Figure 5 and add one classifier at the end (Figure 6). The weights of trained streams are reused to initialize the two branches of the whole network.

5.3 Other accomplishments

Beside data preprocessing and model construction, we have also successfully configured a GPU environment for Keras/Theano to speed up training process. In addition, the paper mentions about using CaffeNet to initialize the network. However, this pretrained network has to be converted to Keras format before loading. We utilize a modified version of Keras that includes conversion package, provided by Bolanos [3].

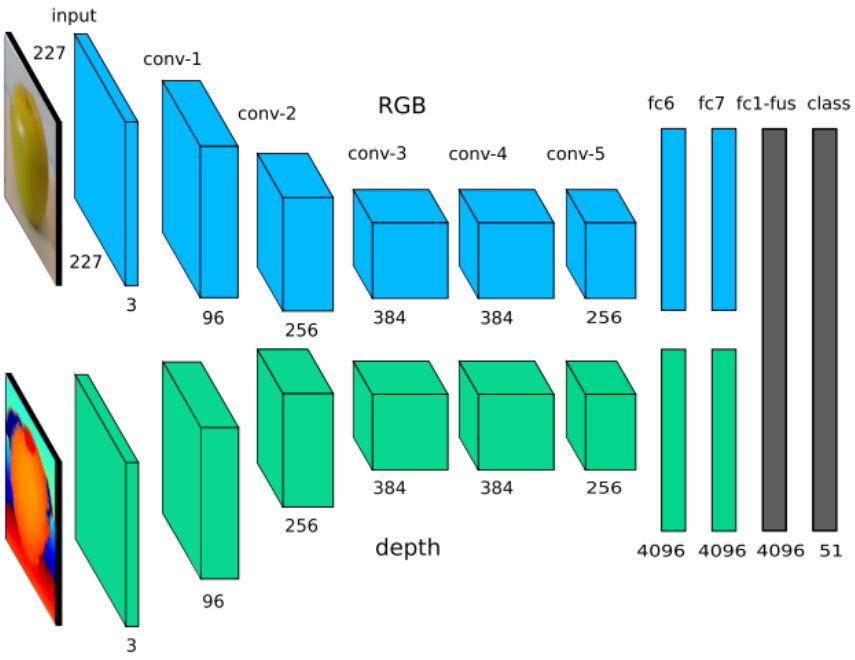


Figure 4: Model architecture proposed by Eitel et al. [1].

6 Current reservations and questions

(if any)

References

[1] A. Eitel, J. T. Springenberg, L. Spinello, M. A. Riedmiller, and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," *CoRR*, vol. abs/1507.06821, 2015.

[2] 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*, pp. 1817–1824, May 2011.

[3] M. Bolanos, "Deep learning library for python. convnets, recurrent neural networks, and more. runs on theano and tensorflow," 2016. [Online; accessed 9-April-2016].

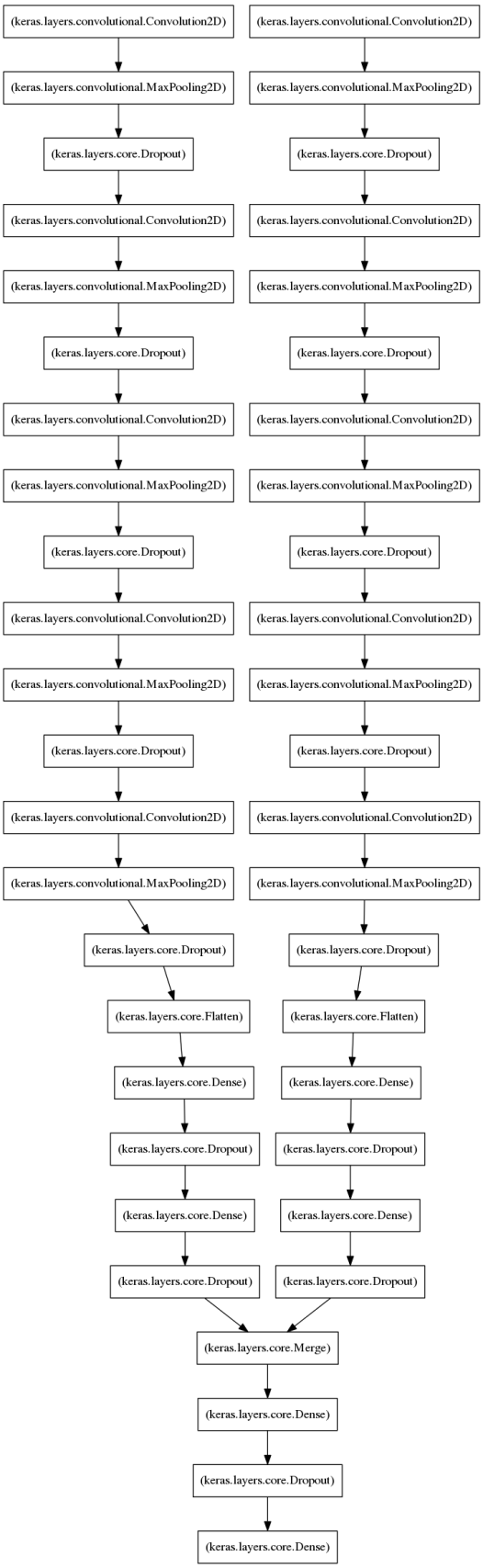


Figure 5: Fusion model with RGB and depth streams.

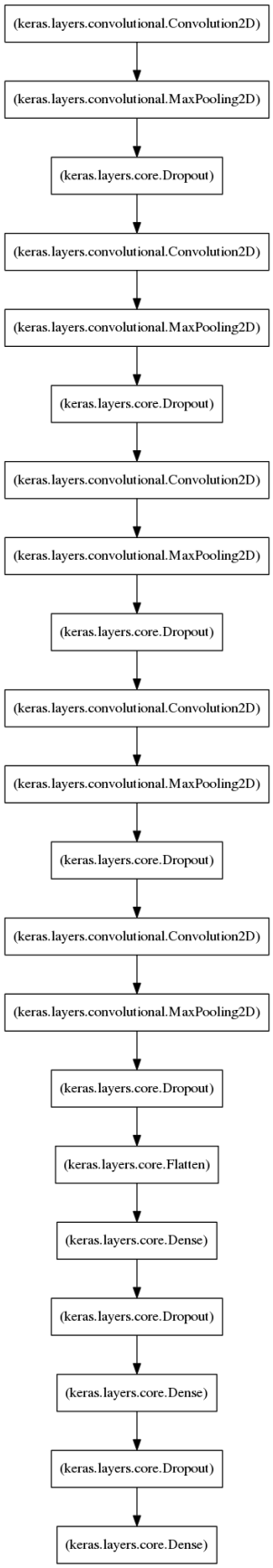


Figure 6: Single stream model.