RGB-D Fusion for Object Detection

Chayan Kumar Patodi
University of Maryland
College Park
ckp1804@umd.edu

Nikhil Mehra
University of Maryland
College Park
nmehra@umd.edu

Raghav Nandwani
University of Maryland
College Park
raghav15@umd.edu

Abstract

In this project we explore, whether the deep fusion technique of depth data along with the RGB data of the image can increase the performance of the current state-of-the-art single-shot detectors for real time object detection e.g. YOLOv2. Depth data can be collected using depth cameras such as a Kinect or stereo setup. We experimented with the deep fusion technique as opposed to the early fusion, late fusion, mid fusion [10] techniques explored by some of the previous works. We use light-weight network architectures so that we can achieve real-time processing using limited computing resources. Our fusion model performs at par with the other fusion techniques in accuracy and localization of the detections with further room for improvement.
Keywords: Deep Fusion, Single-shot detectors, RGB, Depth

1. Introduction

High accuracy is a necessary trait of computer vision applications. An example for such an application is pedestrian localization for autonomous systems. It is highly essential that these systems must not only work accurately, but also in real-time. Some deep learning models such as Fast R-CNN[1], Faster R-CNN[8], YOLOv2[5], YOLOv3[7] are apt for such exacting tasks. However, there are still several issues that need to be solved. For example, detecting small objects, detecting occluded objects are some of the problems that current state-of-the art detection systems are facing. The biggest weakness in the real-time detection systems is their difficulty to provide accurate bounding boxes around the detected objects.

The current state of the art single-pass detection networks like YOLOv2 and SSD(Single-shot detector) predicts output for the bounding box and class scores simultaneously. These types of networks are a lot faster than the other state of the art networks such as RCNN and Faster RCNN. While these state of the art networks have impressed us by their results, there is still some need for improvement. Some previous work has used sensor fusion in order to improve the accuracy of these networks. For example, adding an extra information such as thermal imaging allowed these networks to perform in both day and night scenarios.

The fusion of depth and RGB data with fusion techniques such as early fusion, mid fusion and late fusion are already been successfully. These techniques work by running two different networks for two different data streams and then fusing the data at the end, mid and initial layer respectively. Our baseline paper also worked on the approach of finding an optimal position of fusing the two data streams. It has been found that the fusion in mid-to-late layers in desirable and give good results.

Our main contribution for this project is as follows:- We propose a deep fusion network architecture for the fusing of depth and RGB data, where the fusion of depth and RGB data happens at each consecutive layer.

2. Dataset and Code

We used the EPFL Pedestrian dataset[4]. There are two scenarios in the dataset. The first one (EPFL-LAB) contains around 1000 RGB-D frames with around 3000 annotated people instances. There are at most 4 people who are mostly facing the camera, presumably the scenario for which the Kinect software was fine-tuned. The second one (EPFL-CORRIDOR) was recorded in a more realistic environment, a corridor in a university building. It contains around 3000 frames with up to 8 individuals, split in multiple sequences. We used the open-source code made available by our baseline paper `https://www.gitlab.com/eavise/lightnet`.

3. Related Work

Going through the some state of the art one shot real-time object detection architecture like YOLO [6] and SSD [3]. Being incredibly fast, these architectures shows pretty descent accuracy ImageNet and COCO evaluation metrics. We thought of combining this with data from other sensor to improve the accuracy of these system, which we think will be ideal for robotics application. We look for papers

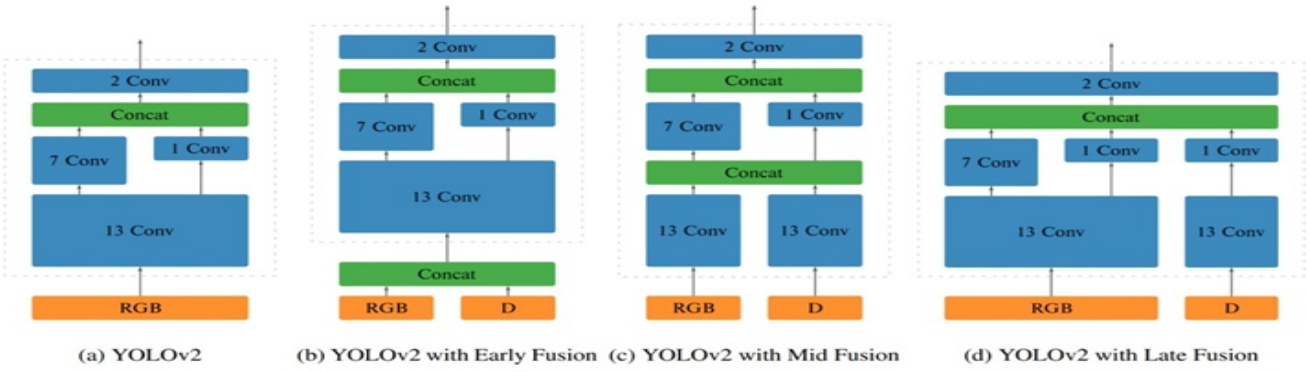

Figure 1. Fusion techniques explored [10]

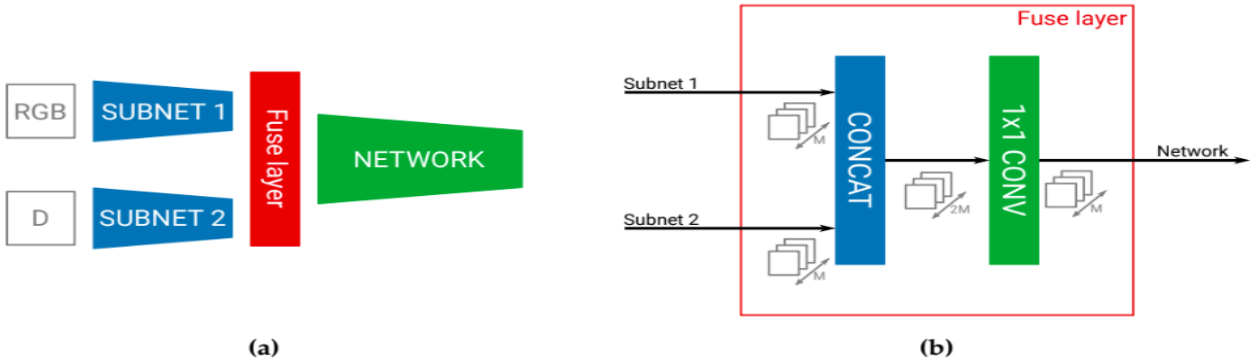

Figure 2. (a) Fusion layer arbitrary inserted in between the original YOLOv2 architecture , (b) Fusion layer concatenate feature from both network and reduce the resulting output to half

that uses sensor fusion with visual data for object detection. The paper we found most appropriate i.e. improving accuracy without significantly compromising with the speed of the output. The paper we found most appropriate was the RGB-D fusion for Real-Time Object detection [9]. After observing the results of this paper we observed that fusion of depth data at any level of the architecture increased the performance compared to just RGB input. The paper also concluded that the fusion in mid to late layer of the network produced best results for e.g. in AP_{COCO} metric the architecture produced best result when fused in the 17^{th} layer. The model performs exhaustive search to decide which layer depth data to be fused to generate best results. At the fuse layer it concatenates the feature map from both RGB and Depth sub network and perform 1x1 convolution to match to the dimension of the next layer and rest architecture is same i.e. m feature maps from RGB subnet and m feature maps from Depth subnet, concatenates generating 2m feature maps and then 1x1 convolution to give a combined m feature maps and hence provide what original number of feature maps which were there in the network. For the training of this model they start from pre-trained weights from ImageNet dataset and performed transfer learning.They compared the results with RGB only and Depth only network as their ground truth.

4. Our Approach

Our project idea revolved around the thought of fusing low level features with their corresponding depths and high level features with their corresponding depths. This idea seems infeasible without a pipeline approach. Also as this project is a fulfilment towards a deep learning course we wanted to explore a deep neural network approach instead of the pipeline approach.

Our deep fusion model(Fig. 3) architecture consists of two different networks(YOLOv2 architecture) for two different data streams namely depth and RGB. As the two networks are of YOLOv2 architecture, these networks consist of a total of 27 layers individually. After each convolution the activation maps from both the networks are concatenated and passed through a 1x1 convolution to reduce the size of the concatenated features. After 1x1 convolutions it is convolved using a filter, pooling and a RELU nonlinearity similar to the YOLOv2 architecture. Input to one layer is a RGB image and input to another layer is a depth map of the same image.

We used the pytorch code as provided by our baseline

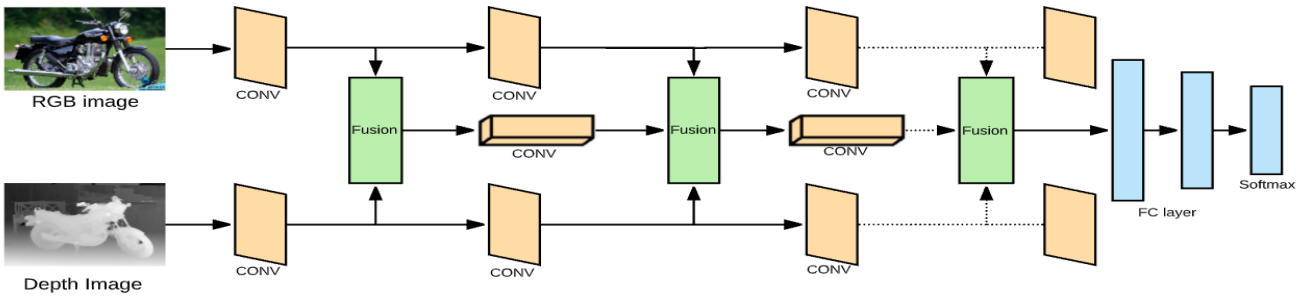

Figure 3. Deep fusion Architecture with fusion at every layer

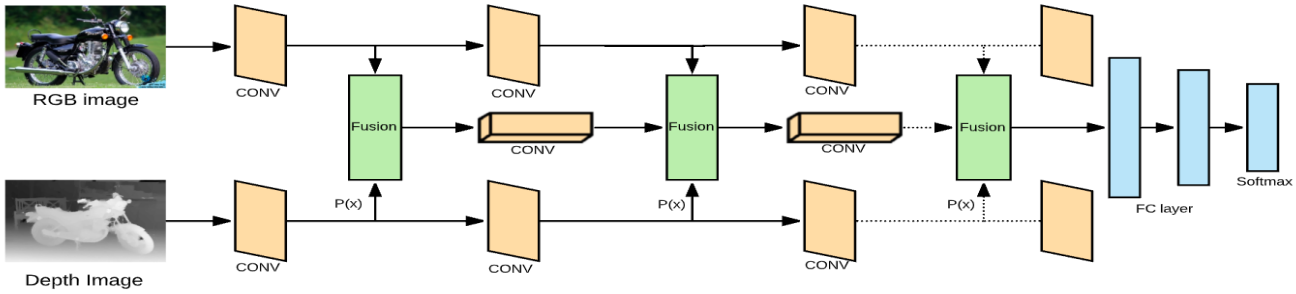

Figure 4. Deep fusion Architecture with probability

paper. The code consists the YOLOv2 architecture with fusion ability. Backpropagation code for propagating parameters to both the network before fusion is also given by the baseline paper. All hyper-parameters are considered similar to the hyper-parameters of the YOLOv2 network.

For training first we tried training the whole architecture end to end. But the solution was not converging and hence we tried a different training approach, wherein we first train our two networks individually. We trained one network using a RGB images and the second network using a depth maps of the corresponding RGB images. Results and evaluations are discussed below.

5. Results

Intersection over Union is a ratio between the intersection and the union of the predicted boxes and the ground truth boxes.the area overlapping the prediction box and ground truth box is the intersection area and the total area spanned is the union.The COCO evaluation metric [2] recommends measurement across various IoU thresholds , which is given by I in our case and have a range of values. The AP summarises the shape of the precision/recall curve, and is defined as the mean precision at a set of equally spaced recall levels. Annotations are the ground truth boxes and Detections as the name suggests are detection.

$$AP_{COCO} = \frac{\Sigma_{IoU \in I} AP_{IoU}(Annotations, Detections)}{I};$$

$$I = \{0.50, 0.55, 0.60, .., 0.95\}$$

We performed exhaustive experiments on our application case , The first application case is evaluated using a dataset which consists of Kinect RGBD images of people.

Our model performed at par with the baseline. We did not see any improvement in the accuracy as compared to the baseline for the EPFL dataset. Average precision results of the baseline and our model is shown in the table below.

	Average Precision (AP_{COCO})(%)
YOLOv2	53%
Fusion in 17^{th} layer (Baseline)	55.23%
Fusion in 18^{th} layer (Baseline)	56.57%
Deep Fusion (Our Model)	55.4%

6. Conclusion and discussion

At the starting of the project, Raghav and Chayan worked on data preparation and evaluating ground truth for the project, while Nikhil worked on modelling and implementing the fusion as proposed in the project architecture meanwhile referring to the architecture LightNet [9]. Then we all together worked collectively on writing the code for loading the dataset along with that training and evaluation of our model. There was problem in data preparation as the LightNet takes ".h5" data file as input and we had separate RGB and Depth images in EPFL dataset [4]. Then we first converted the RGB and Depth images dataset in .h5 file

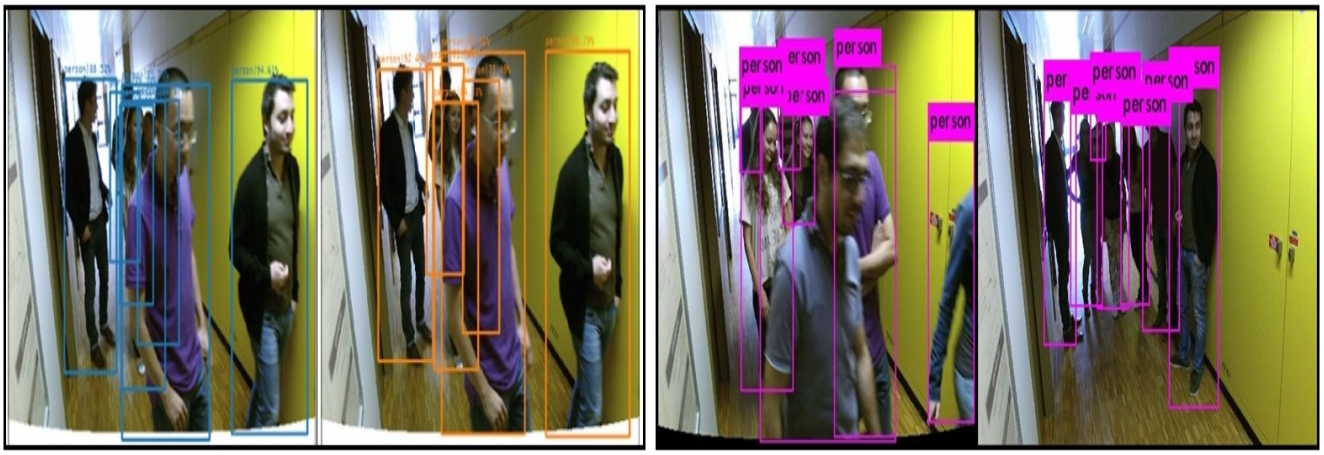

Figure 5. Baseline Results (2 Images on the left) vs Detection Results(2 Images on the Right)

meanwhile dividing the dataset in 70:10:20 ratio for training,validation and testing. We faced difficulty in training the architecture as the model was too complex.

7. Future Work

Further we want to explore a architecture where the fusion is done based on the random probability assigned to that layer (Fig.4) during training for each epoch, an idea similar to dropout.

References

[1] Girshick. Fast r-cnn. *arXiv*, 2015.

[2] Belongie Perona Ramanan Dollár Zitnick Lin, Maire. Microsoft coco: Common objects in context. in proceedings of the european conference on computer vision (eccv), 2014.

[3] Erhan Liu, Anguelov and Szegedy. Ssd: Single shot multibox detector. *arXiv*, 2016.

[4] Swiss Federal Institute of Technology Lausanne. Epfl rgb-d pedestrian dataset. *https://www.epfl.ch/labs/cvlab/data/data-rgbd-pedestrian/*, 2012.

[5] Redmon and Farhadi. Yolo9000: Better, faster, stronge. *arXiv*, 2017.

[6] Girshick Redmon, Divvala and Farhadi. You only look once: Unified, real-time object detection. *arXiv*, 2016.

[7] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[8] Girshick Ren, He and Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv*, 2016.

[9] T.Goedeme T. Ophoff, K. V.Beeck. Exploring rgb+depth fusion for real-time object detection, 2018.

[10] T.Goedeme T. Ophoff, K. V.Beeck. Improving real-time pedestrian detectors with rgb+depth fusion, 2018.