Weakly Supervised Object Localization on grocery shelves using simple FCN and Synthetic Dataset

Srikrishna Varadarajan*, Muktabh Mayank Srivastava⋆

Paralleldots, Inc.⋆⋆⋆

Abstract. We propose a weakly supervised method using two algorithms to predict object bounding boxes given only an image classification dataset. First algorithm is a simple Fully Convolutional Network (FCN) trained to classify object instances. We use the property of FCN to return a mask for images larger than training images to get a primary output segmentation mask during test time by passing an image pyramid to it. We enhance the FCN output mask into final output bounding boxes by a Convolutional Encoder-Decoder (ConvAE) viz. the second algorithm. ConvAE is trained to localize objects on an artificially generated dataset of output segmentation masks. We demonstrate the effectiveness of this method in localizing objects in grocery shelves where annotating data for object detection is hard due to variety of objects. This method can be extended to any problem domain where collecting images of objects is easy and annotating their coordinates is hard.

Keywords: object localization, weak supervision, FCN, synthetic dataset, grocery shelf object detection, shelf monitoring

1 Introduction

Visual retail audit or shelf monitoring is an upcoming area where computer vision algorithms can be used to create automated system for recognition, localization, tracking and further analysis of products on retail shelves. Our work focuses on the task of product localization on shelves, i.e object detection. We approach the task as weakly supervised object detection and our method ends up solving the task of product image recognition (classification) too as a subtask.

Manual annotation of object bounding boxes across images having a large number of instances per image and/or a large number of object categories is both a time consuming and an extremely costly process. One can observe that these numbers are low for most publicly available object detection datasets (average instances per image is less than 3 for PASCAL VOC and ImageNet and 7.7 for MS-COCO, number of classes is 20 for PASCAL VOC, 200 for ImageNet, 80 for MS-COCO). Annotating retail product bounding boxes in images is infeasible

⋆⋆⋆ `https://paralleldots.xyz`

⋆ Authors contributed equally

due to number of retail products available, frequent changes in packaging, introduction of new products (large number of classes and dynamic appearance) and high density with which objects are placed in shelves (large number of instances per image). This difficulty in acquiring annotated dataset limits the use of fully supervised object detectors in such datasets.

To perform object detection tasks in such datasets, we need a method which uses just object instances and not only learns to recognize the instances, but also localize them. Our work proposes a method for the same. Methods like ours are classified into weakly supervised object detection methods.

We aim to do object localization using the same network which is trained for the task of classification on object instances. The classifier we use here is a Fully Convolutional Network (FCN). All the learnable layers in FCN are convolutional layers. There are no fully connected (dense) layers in the network. This allows us to pass an input of any higher resolution to the network and get a corresponding output mask. We also make use of another optional network to enhance our outputs from the FCN for our object detection task. For this network (Refinenet), we use a convolutional network based on the encoder-decoder framework which we call as ConvAE. ConvAE is trained on an artificial dataset created using the object instances. The output of the ConvAE is converted to obtain bounding boxes for the object detection task. Please note that we use object detection and object localization invariably throughout this paper.

2 Weak Supervision and Synthetic Datasets

Weakly supervised learning is a method of learning in which the supervision can be indirect, inexact and incomplete. In this setting, one doesn't have access to supervised data but some other form of labeled data which is useful. There have been previous works [1] [2] [3] [4] [5] which does the weakly supervised localization (WSL) task on the standard datasets like PASCAL VOC and COCO. The WSL methods are usually Multiple Instance Learning (MIL) based or end-to-end CNN based networks. Our method falls into the latter category.

The FCN component of our method shares some conceptual similarity with [3], where a classifier trained on instances is used for object detection too. Our model is simpler and doesn't use multi-scale sliding windows. Our model also gives perfect bounding boxes unlike the point (center) output from latter. There is also difference amongst the datasets both the algorithms are trained/tested on, while we train it on single label object images for retail objects, the latter algorithm has been trained on multilabel datasets like tags of PASCAL VOC.

Our optional refinement module is trained on the Synthetic Dataset. The creation of Synthetic Dataset has been explored in the past for various object detection tasks [6] [7] [8]. Some use sophisticated methods to place the objects in its environments to expand an existing dataset or create a new one, while we just simply place objects on empty shelves with few logical parameters for randomness. The novelty of our approach is that the optional refinement module is not trained on directly synthesized images of Synthetic Dataset, but rather their rep-

resentation after passing through FCN. Moreover, we use the Synthetic Dataset only to improve the extent of our localization and to reduce false positives.

3 Method

3.1 FCN

FCN: Training During training task, we perform classification on instances of the objects. The network architecture of the FCN is as follows. We take the pretrained layers of VGG11[9] from torchvision[1] and remove all the FC Layers. We then add a single convolutional layer at the end with appropriate kernel size so as to get a Nx1x1 output, where N is number of classes. This is shown in Figure 1.

Fig. 1. Classification Pipeline on object instances

FCN: Testing One of the core advantages of our method is that FCNs are independent of input size. We take the valid assumption that the object instances (training images) are smaller in size than testing images (images containing multiple objects). Hence, the FCN which is designed to do a classification task (1x1 output) will give a corresponding output mask when a higher resolution input is passed.

In the testing task, we pass an image pyramid (consisting of various downscales of the image) to the same network. We use downscale factor of 1.5 to generate the pyramid. We resize the corresponding outputs from each level of pyramid to the original image size. We then take the mean of the pyramid outputs as our final output mask as shown in the Figure 2.

Since the network is trained on individual instances of the object, it's possible that the dataset doesn't capture the various sizes of the object. This can be a huge drawback during test time. But this is overcome by the image pyramid method employed during test time. The idea is that network would be able to capture the object at least at one downsampled level.

[1] `https://github.com/pytorch/vision/blob/master/torchvision/models/vgg.py`

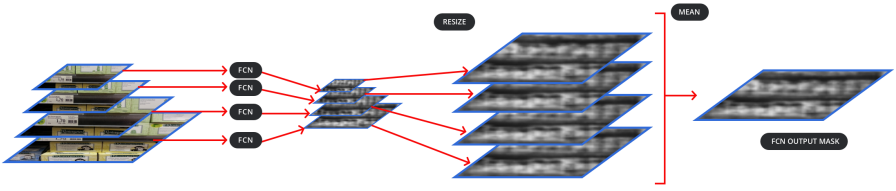

Fig. 2. Object detection pipeline on shelf images

3.2 Refine-Net

Refine-Net module is used for refining the outputs of FCN during object detection task. FCN output maps capture the objects, but they focus on more discriminative areas as they were trained for classification. This can localize the discriminative features of objects, but cannot always give bounding boxes surrounding them. As a result it might give low IoU scores. FCNs also may have false positives on background (due to lack of background training data). Refine-Net is hence used to address both these limitations. It is convolutional network which is trained to improve the output maps generated by the FCN. Please note this is an optional module and is not needed in case FCN works well on its own. One can see the performance differences later in the results section.

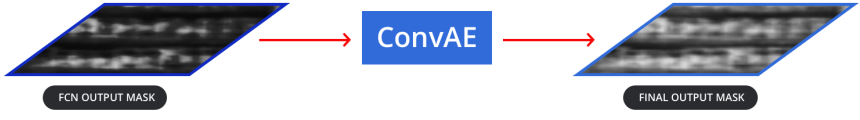

Fig. 3. Refinement Module

Synthetic Dataset For this task, we create a synthetic dataset using the object instances from the trainset. We randomly take few images (20-50) from Google Images using the keyword *empty shelves*. On these empty shelf images, we randomly place objects in a planogram format with varying number of columns

(within each image) and varying number of rows (across images). We also scale the object sizes randomly between logical values. Eg: 0.5-1.1. We also tried this with a black background instead of empty shelves. We see a small increase in the mAP when we use the empty shelves.

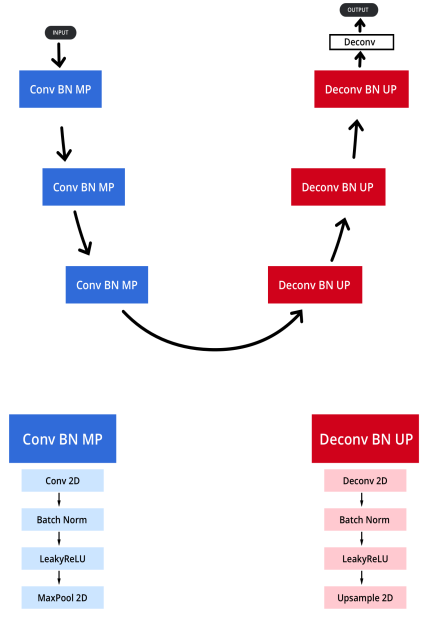

Fig. 4. Refine-net architecture

Refine-Net: Architecture We name the architecture used as Refine-Net as ConvAE. The ConvAE network follows the standard Encoder-Decoder framework with few layers of Convolutional and Downsampling layers followed by equivalent number of Deconvolutional and Upsampling layers. All the Conv2D and Deconv2D layers have 3x3 kernel size except the final Deconv which is 1x1. All MaxPool2D layers have kernel size 2x2 while all the Upsample layers are bilinear and have factor as 2. The architecture is illustrated in the Figure 4.

Refine-Net: Method During training, the images from synthetic dataset are passed through the FCN and the segmentation output is obtained. The FCN output mask is the input to the ConvAE (Refine-Net). We train the ConvAE with CrossEntropyLoss and ground-truth which is known to us from the Synthetic Dataset.

During testing, the output of the FCN is passed through ConvAE to be enhanced and made better for localization. The output segmentation mask of ConvAE is now processed for extracting bounding boxes. We first binarize the mask by applying a threshold. Then, extract the bounding box coordinates from the connected components of the binarized mask. This process is shown in Figure 5.

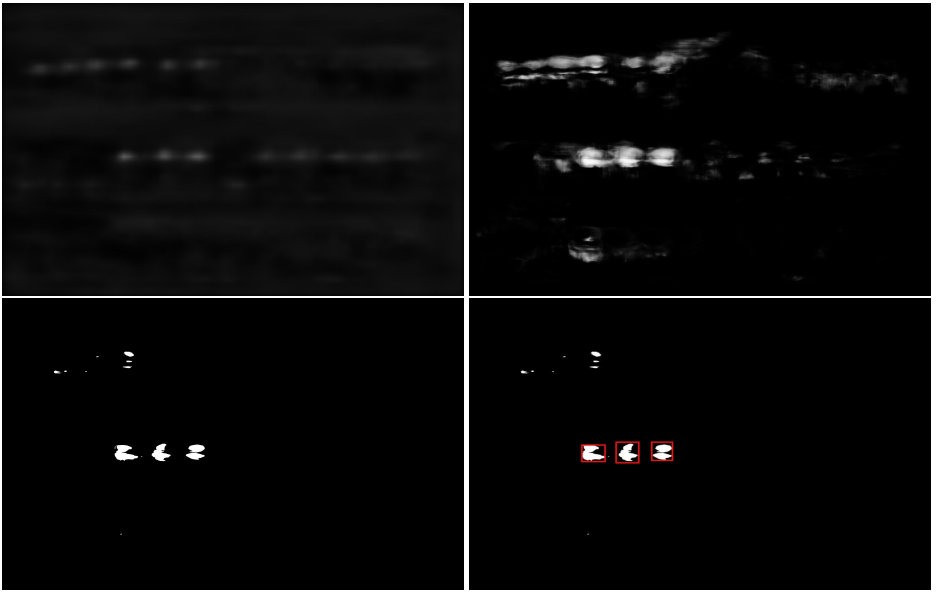

Fig. 5. From top left (clockwise): Output from FCN, Output from ConvAE, Output after thresholding, Output after creating bounding boxes

3.3 Observation

In the datasets of product recognition, it often happens that there are very similar products which belong to different classes. This tends to make the classifier's (FCN) output probability of the correct class a little weaker. Here, the confidence score of the correct class isn't as high as we'd like it to be. In such cases, one can find reasonable confidence score in **other classes** even though the FCN predicted the correct class[2]. Hence, this induces a lot of false-positives depending on the **threshold** of converting the segmentation into a binary mask.

The ConvAE works great in eliminating these false-positives. It is able to learn the prominent class and remove the false-positives from other classes easily.

[2] gave maximum probability for the correct class

Hence we don't have to worry about the threshold[3]. We note that it's not possible to train the ConvAE if our classifier isn't strong. The false positives makes it very difficult for the network to learn the refinement task.

4 Experiments

We were interested in applying this method to a grocery shelf dataset. We also wanted to test the performance of the method in detecting different products from the same brand which look very similar. We were able to find one such public dataset [10] which is most similar to our objective, we're calling it the Cigarette Dataset. We also wanted to show the performance of the method in a dataset which is not small and controlled but bigger and more generalized. Hence, we chose the Grocery Dataset [11]. The dataset details, experiments and results on both the datasets are explained in the following sections

5 Cigarette Dataset

5.1 Dataset description

It's important to explain the dataset to understand our experiments and results. This dataset consists of various products from multiple brands of cigarettes. The dataset considers 10 brands as positive classes and the rest as background. The dataset consist of 3 parts

1. Product Images
2. Brand Images
3. Shelf Images

Product Images are the instances of products of each brand. It contains 5 unique product images taken by 4 different cameras in different lighting, angles and noisy versions making the total number to be around 350 for each product of a brand. **Brand Images** are taken directly from the Product Images by cropping out only the brand logos from the cigarette boxes. There are 354 **Shelf Images** consisting of grocery shelf images. It contains various cigarette product boxes from different brands. There are 13,000 products in Shelf Images overall and all of them are annotated. Around 3000 belong to the 10 classes, the rest 10,000 products are considered as negative or background class.

Negative object instances contains products which don't belong to the 10 brands as well as products which belong to the 10 brands but are not the products in our 10 classes. This is explained in Figure 6. Chesterfield (brand) doesn't belong to any of the 10 brands and is a negative class while Marlboro (brand) is one of the 10 brands but the given instance is a different product than those in our trainset. Hence, this is also a negative class. A positive example of Marlboro class can be seen in Figure 7.

[3] Note that in this case, the classifier predicts the correct class, just the confidence score probabilities at the spatial location is lesser

Fig. 6. Negative examples: Belongs to background class

Fig. 7. Positive example: Belongs to Marlbaro class

5.2 Train set

The training task is the classification task. For our experiment, we chose the Brand Images as the trainset to do the classification with our FCN. Using the Product Images induces lot of false positives. The Product Images have two parts - the brand logo and the warning labels. The former is the only discriminative part as the latter is common to all brands. ¡¡Since our dataset contains only 5 unique products, that information isn't reflected in the trainset¿¿. The network is made to think that certain warning labels are specific to each product. This is the reason why using Product Images results in lot of false positives. Hence, we chose to use the Brand Images instead.

For background patches, we took a very small amount of (6.49%) images from the Shelf Images, containing all 11 images from one shelf and all 12 images from a different shelf. We make sure that there's no overlap in the shelves in our train and test set. The exact details are given in Appendix. From these images, we extract background patches which used as the negative class in our classification task. We randomly use 20% of this trainset to be our validation set.

A synthetic dataset is created to train RefineNet ConvAE by using product images in train set and following procedure detailed earlier.

5.3 Test set

The testing task is the object detection task. The Shelf Images excluding the ones used for background patches, are used as the test set. This comes around to 331 test images. The annotations are provided for all instances of the products in each shelf image.

5.4 Previous Work

An initial work on this dataset is done by [10]. They first try to segment all the products by calculating number of shelves and applying a height and width constraint from the dataset. This works well in segmenting all the boxes as the shape of products in this dataset is rather constant. They also classify each product image into given classes. But they do not combine both of them to give a working object detection method.

We don't have any other previous work to compare with, so we make a simple baseline which is described as follows. We employ a sliding window of different scales, aspect ratios on the test image and classify the sliding window into different classes. We then use non max suppression (NMS) to remove the redundant bounding boxes. The classifier used here is the same FCN which is used in other results as well. This baseline would show the significance of the image pyramid testing methodology as well as the significance of ConvAE.

5.5 Implementation Details

For the **FCN**, we set the last convolutional layer kernel size as (2,4). We train the FCN using SGD with momentum 0.9 and learning rate 1e-3. We use weight decay of 5e-4 and an early stopping of 30 epochs. For creating the **Synthetic Dataset**, we set the image height and width as (2000,3000) and (1200,2000) according to number of shelves (rows) in the image. For the **ConvAE**, we set the number of filters in the 3 conv and deconv blocks as 16,24,32 respectively. We train the ConvAE using SGD with momentum 0.9 and learning rate 1e-3. We use weight decay of 5e-4 and early stopping of 5 epochs.

5.6 Results

The mAP of different models are shown in Table 1. The mAP is calculated as the mean of Average Precision (AP) of all classes. The calculation of AP is taken directly from VOC devkit (Matlab[4], Python[5]). We use the VOC07 11 point metric.

Our image pyramid method performs better than the baseline sliding window method. The results get better when we add the refinement module to the FCN

[4] `http://host.robots.ox.ac.uk/pascal/VOC/voc2012/#devkit`

[5] `https://github.com/rbgirshick/py-faster-rcnn/blob/master/lib/datasets/voc_eval.py`

Table 1. Object Detection results on Cigarette Dataset

Model	mAP (iou=0.1)
Selective Search (with FCN Classifier)	-
Sliding Window (with FCN Classifier)	0.278
Pyramid FCN Model without AE	0.2805
Pyramid FCN Model with ConvAE	0.3069

model (Table 1). The difference in accuracy is less due to the fixed structure of objects in the cigarette dataset. This gives the sliding window method an extra knowledge and allows it to get a decent accuracy. We show later in Section 6.4 that this method fails badly in case of a varied dataset which has different object shapes and sizes, while our image pyramid method works much better[6].

Fig. 8. Sample Results of Pyramid FCN Model with ConvAE. Blue shows ground-truth boxes while Red shows model's prediction

We also tried to compare about method with another baseline by taking region proposals from selective search [12] and classifying it with our classifier. But performing the selective search on each image took around 10 minutes and would've taken almost 3 days to test on the entire testset. Hence we tried it on a subset. We saw very poor performance, in the order of 1e-4 mAP. Hence, we decided not to test on the entire dataset. The aim was to compare against the classic region proposal methods [13]. Note that we can't use fully supervised methods like RPN [14].

6 Grocery Dataset

6.1 Dataset Description

A supermarket dataset containing 8350 training images spanning 80 semantic object categories was introduced by [11]. The training images are taken under

[6] Note that these are absolute values, not percentages.

ideal, studio conditions. Their test set contains 680 real shelf images taken in different grocery stores.

The annotation on the test set covers a group of individual products **as opposed to** giving bounding boxes for each instance following the object detection task. Fortunately, [15] released the bounding box annotation for a subset (74 images) of the test set. We use this as our test set. The comparison between the annotations is shown in 9. This subset contains 12 classes. Hence, we train our FCN and the ConvAE only on these 12 classes. We observe a similar trend on the full testset but are unable to measure it due to lack of annotations.

Fig. 9. Annotations from [11] (left) and [15] (right)

6.2 Previous Work

A method was proposed by [11] to recognize products on this dataset, but it doesn't perform the object detection task that we aim to do. Their method consists of 3 steps: Multi-class ranking, Fast Dense Pixel Matching and a genetic algorithm based multi-label image classification. The evaluation results shown based on the group level annotations shown in Figure 9 (Left). [15] have also worked on the same dataset, but their focus was on planogram compliance. They're given the structure of a planogram (called reference planogram) along with the objects it contains and are asked to check whether the observed planogram is compliant to the planned (reference) one. They also detect missing or misplaced items. We are unable to find any suitable previous benchmark, so we follow the same baseline as described in Section 5.6

6.3 Implementation Details

For the FCN, we set the last convolutional layer kernel size as (7,5). The input image is of fixed dimensions taken from average of the dataset. We train the FCN using SGD with momentum 0.9 and learning rate 1e-3. We use weight decay of 5e-4 and an early stopping of 30 epochs. We used common data augmentation techniques like flipping, scaling, rotations using imgaug package[7]

We tried using the Refine-net here, but it wasn't helpful as the FCN isn't very strong in generating the feature maps without false-positives. This is mentioned in Section 3.3. So, we report the results directly on outputs of FCN, which is thresholded and converted to bounding boxes (Figure 5 without the refinement step).

6.4 Results

We follow the same evaluation metric as described above in Section 5.6. As mentioned in Section 5.6, one can see that the sliding window method with multiple scales, aspect ratios and NMS fails to work in the case where the objects don't have a fixed aspect ratio or size. For selective search model, we face a similar problem as well as similar performance (explained in Section 5.6). The image pyramid testing method using our FCN works much better than the baseline (Table 2). The results show that the method works well even for generalized datasets which has semantic grocery categories containing only one or few instances of each product.

Table 2. Object Detection results on Grocery Dataset

Model	mAP (iou=0.1)
Selective Search (with FCN Classifier)	-
Sliding Window (with FCN Classifier)	0.0249
Pyramid FCN Model without AE	0.2801

6.5 GroZi-120 dataset

Grozi-120 dataset [16] contains 120 specific product instances as training set. The test-set contains 29 videos of shelves containing these 120 products. [11] had selected 885 frames from these 29 videos and released the annotations to show their results. Unfortunately these annotations aren't bounding boxes but just list of classes present in the frame. It doesn't give the bounding boxes of all the objects present in the frame. Hence, we were unable to show the results on this dataset.

[7] `https://github.com/aleju/imgaug`

Fig. 10. Sample Results of Pyramid FCN Model without AE. Blue shows ground-truth boxes while Red shows model's prediction

6.6 Discussion

We note that the IoU threshold for calculating the mAP is low (0.1), this is due to two factors. First, the weakly supervised method is unable to fit the entire object. Second, it gives a combined bounding box in the cases where the boundary between similar objects is very hard to distinguish. The latter is unique to shelf datasets and shouldn't be a limitation of the method in general datasets. The former can be tackled as well. Our refinement module is complementary to other methods like [5] which helps to improve the extent of the localization and we expect our method's performance to increase when combined with such methods. Future work can be focused on improving the method's performance on higher thresholds.

7 Conclusion

We have proposed a novel and extremely simple method to do object localization using just the object instances (classification labels). We have also proposed an optional refinement module to increase the extent of localization and to reduce false positives as well. We have implemented it on relevant datasets and shown good performance on all of them. We've also established the superior performance of the method on a varied and general dataset as well.

References

1. Li, D., Huang, J.B., Li, Y., Wang, S., Yang, M.H.: Weakly supervised object localization with progressive domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016)
2. Jie, Z., Wei, Y., Jin, X., Feng, J., Liu, W.: Deep self-taught learning for weakly supervised object localization. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 4294–4302
3. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free? - weakly-supervised learning with convolutional neural networks. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 685–694

4. Teh, E.W., Rochan, M., Wang, Y.: Attention networks for weakly supervised object localization. In: BMVC. (2016)
5. Singh, K.K., Lee, Y.J.: Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. 2017 IEEE International Conference on Computer Vision (ICCV) (2017) 3544–3553
6. Georgakis, G., Mousavian, A., Berg, A.C., Kosecka, J.: Synthesizing training data for object detection in indoor scenes. CoRR **abs/1702.07836** (2017)
7. Su, H., Zhu, X., Gong, S.: Deep learning logo detection with data expansion by synthesising context. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV) (2017) 530–539
8. Tian, Y., Li, X., Wang, K., Wang, F.Y.: Training and testing object detectors with virtual images. CoRR **abs/1712.08470** (2017)
9. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
10. Gl Varol, R.S.K.: Toward retail product recognition on grocery shelves (2015)
11. George, M., Floerkemeier, C.: Recognizing products: A per-exemplar multi-label image classification approach. In Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., eds.: Computer Vision – ECCV 2014, Cham, Springer International Publishing (2014) 440–455
12. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. International Journal of Computer Vision **104** (2013) 154–171
13. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014) 580–587
14. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2015) 1137–1149
15. Tonioni, A., di Stefano, L.: Product recognition in store shelves as a sub-graph isomorphism problem. CoRR **abs/1707.08378** (2017)
16. Merler, M., Galleguillos, C., Belongie, S.: Recognizing groceries in situ using in vitro training data. 2007 IEEE Conference on Computer Vision and Pattern Recognition (2007)

8 Appendix

ShelfImages used in training for background patches:

1. CigaretteDataset/ShelfImages/C1_P01_*.JPG (11 images)
2. CigaretteDataset/ShelfImages/C2_P01_*.JPG (22 images)