Deep Neural Networks for Object Enumeration

Zihao Xu
Department of Computer Science
Pomona College
Claremont, CA
zxql2015@MyMail.pomona.edu

Mariam Salloum
Department of Computer Science and Engineering
UC Riverside
Riverside, CA
msalloum@cs.ucr.edu

***Abstract*—Estimating object count from images is a difficult problem that has a wide range of applications. In this work, we examine the object counting problem for images from the Amazon Bin Images Dataset. This task is riddled with many challenges, including occasional low image quality, object occlusions, and diversity in objects. This work explores a deep-learning approach using a CNN architecture for this object counting problem. Our solution combines end-to-end training on ResNet with test time augmentation, achieving promising results for this difficult task.**

***Keywords*-Convolutional Neural Network (CNN); Amazon Bin Images Dataset; ResNet; Object Counting**

I. Introduction

Convolutional neural networks (CNNs) have been leveraged to successfully tackle a number of tasks, including image classification [1], and image segmentation [2]. In this paper, we study the object counting problem, which aims to estimate the number of objects in a still image. Specifically, we examine this problem for images from the Amazon Bin Image Dataset (ABID [3]), which depict objects within pods in an operating Amazon Fulfillment Center. Figure 1 shows sample images from the dataset. We present and compare several deep-learning approaches to solve this problem.

The paper is organized as follows. Section II presents related work on the object counting problem in various applications. Section III provides a high-level description of the dataset and presents the problem formulation. Section IV describes our steps in training multiple variations of the ResNet [4] model, while Section V provides experimental results. Finally, Section VI concludes this work and offers future directions.

II. Related Work

To our best knowledge, this work is the first to tackle the object counting problem on the Amazon Bin Images Dataset. Previous work has investigated the counting problem in the context of "crowd counting", which aims to count the number of people in a crowded scene [5].

Over the past few years, researchers have presented a variety of approaches for the crowd counting problem. Initial works developed models using handcrafted features, while more recent works use Convolutional Neural Network (CNN) based approaches due to enhanced performance. A complete survey of the crowd counting problem is presented by [6].

III. Problem Overview

In this section, we provide an exploration of the Amazon Bin Image Dataset and the problem formulation.

A. Dataset Description

The dataset consists of approximately 530,000 images with corresponding meta data. The meta data provides the number of objects in the image along with an itemized list of the objects in the image. Note, the meta-data does not contain localization information. Table I presents a description of the dataset.

Num. of images	536,432
Num. of images (object $\leq$ 5)	361,967
Max items in image	209
Min items in image	0
Avg items in image	5.1

Table I: Amazon Bin Images Dataset Description

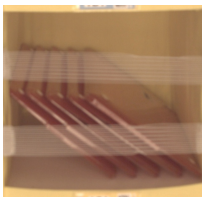
B. Problem Formulation

Given that the final prediction for each image should be a positive integer, the CNN architecture cannot directly model this task. In our work, we develop multiple ways to adapt CNN models to perform integer prediction.

The first approach is to view this problem as a multi-category classification, where each image is classified by the number of items it contains.

The second method is to treat this problem as a regression task, i.e. the trained model will predict a single integer as the number of items in an image. However, one potential problem is that the output of a regression CNN model using linear activation at its last fully-connected layer is only able to output a floating point number, not an integer, as its prediction. We propose two methods to address this problem:

- Round the predictions to the nearest integer.
- Use a loss function that closely models the difference between the *rounded* prediction and the true label.



(a) Ex 1

(b) Ex 2

(c) Ex 3

Figure 1: Figures a-c are examples from the Amazon Bin Images dataset. All three examples (a-c) depict images with 5 items. Note, most images have one or two strips of tape over the shipment box that occludes the items in the image.

In this paper, we explore both classification and regression methods for the counting problem.

C. Potential Challenges

The image counting problem, a particularly difficult task, is rendered more challenging by nature of the Amazon Bin Image Dataset, as described below:

1) A variety of items with different shape, size, and color exist in the dataset, which makes generalization hard for any model.
2) Items within the bins might be partially or completely occluded by other items or the tape that Amazons uses to prevent items from falling.
3) Packages and bundle deals cause confusion to the algorithm's training and prediction processes.
4) Mislabeling of images might exist in the data set.
5) Some images are blurred or smeared.

All these factors contribute to the difficulty in model training and generalization to unseen images.

IV. Approach

In this section, we begin by describing the approach we take for images pre-processing and the set up of training, validation, and test datasets. Then, we will cover two data augmentation techniques employed in training and testing. Lastly, we will discuss the final model selected - end-to-end learning on a classification ResNet using test time augmentation.

A. Image Pre-processing

For consistency, we first re-size the images to $(224, 224, 3)$. To train the ResNet model, we use its implementation in Keras [7] and standardize the images using the mean and standard deviation for each channel.

The dataset is split into training, validation, and test sets, each containing 288000, 36100, 36100 images. Note that due to the sheer difficulty of this task, we only focus on images with less than or equal to 5 objects.

B. Data Augmentation

In this work, we employ two categories of data augmentation techniques: training time augmentation and test time augmentation.

During training time, we randomly augment the images by a horizontal or vertical flip, or a random rotation of the images by 90, 180, or 270 degrees. Due to the large size of the training set, we overwrite the original image if any transformation is performed. Furthermore, when testing the model, we employ test time augmentation by feeding three version of the same image - the original copy, the horizontally flipped, and the vertically flipped copies - to the model, and combine the predictions. In the classification case, we sum up the outputs after the *softmax* operation and then take the *argmax* of the prediction vector to form the final prediction; in a regression case, we average the 3 predictions and round the result.

Section V will demonstrate the effectiveness of both methods in more detail.

C. Training Setup

For the classification model, we adopt the *cross-entropy* loss function to maximize prediction accuracy. For regression models, we test two different loss functions - *Mean Squared Error (MSE)* and *logcoh*. The *logcoh* function more closely models rounded predictions because *logcoh(x)* is approximately equal to $(x^2)/2$ for small x values and $abs(x) - log(2)$ for large x values. As such, it would penalize small errors (± 1) much less than does the standard *MSE* loss function.

During training, we feed images in batches of 32 and compile models using the *Adam* optimizer [8] with a learning rate of $5 * 10^{-3}$. We set number of epochs to 100 and stop training if the validation loss does not decrease in 10 consecutive epochs. Besides, we only save the best model according to validation loss. For all models, we randomly initialize model weights and perform end-to-end training.

V. Experimental Results

In this section, we present 4 models and compare their performances. Table II shows a summary of model specifications and their test accuracy and test *MSE*.

	Specification	Test Acc.	Test MSE
resnet_reg	ResNet + regression	48.21%	0.7972
resnet_reg_logcosh	ResNet + regression + logcosh loss	49.99%	0.8031
resnet_clf	ResNet + classification	53.38%	0.9441
resnet_clf_TA	ResNet + classification + test-time augment	54.64%	0.8982

Table II: Model performances - Test Accuracy and Test *MSE*

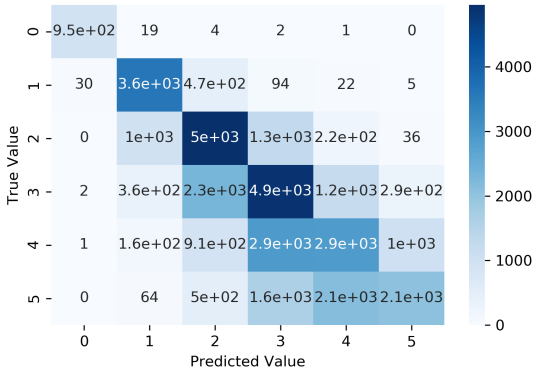


Figure 2: Confusion Matrix for *resnet_clf_TA*

We first trained two base models, the classification ResNet (*resnet_clf*) and the regression ResNet (*resnet_reg*), with and without training time augmentation. In both cases, data augmentation was effective in boosting model performance, and thus we only display models using training time augmentation. Table II shows that the classification model performed better than the regression model in terms of test accuracy, though the regression base model produced lower test *MSE*. We also trained *resnet_reg_logcosh* to explore another loss function for the regression model. Its test accuracy improved slightly over *resnet_reg*, but was still worse than that of *resnet_clf*. Thus, we proceeded by applying test-time augmentation on the classification model to build *resnet_clf_TA*, further improving both the test accuracy and test *MSE*. We present the confusion matrix of *resnet_clf_TA* in Figure 2.

Further, to understand how the model works, we randomly sampled and plotted activation-maximized filters from the trained *resnet_clf* model in Figure 3. These figures show that the trained model recognizes shapes and textures of objects within bin images. We also plotted the correlation between activation values and object counts in Figure 4. These figures show that filters are trained to capture "typical scenes" of bin images with specific number of objects.

VI. Conclusion

In this project, we explored the object counting problem for images in the ABID. We experimented with two kinds of models, regression and classification ResNet, and incorporated test time augmentation to arrive at *resnet_clf_TA*, achieving promising results. Also, we visualized filters within the trained *resnet_clf* and explored the relationship between image labels and layer activation values. In the future, we plan to examine boosting methods that leverage active deep features.

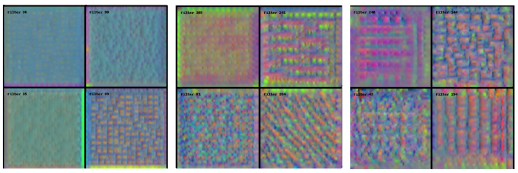


Figure 3: Activation Maximization for Sample Filters

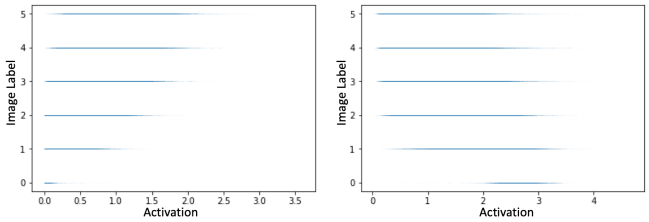


Figure 4: Relationship between Labels and Activation Values

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[2] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014. [Online]. Available: http://arxiv.org/abs/1411.4038

[3] Amazon bin images dataaset. [Online]. Available: https://registry.opendata.aws/amazon-bin-imagery/

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[5] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 833–841.

[6] V. Sindagi and V. M. Patel, "A survey of recent advances in cnn-based single image crowd counting and density estimation," *CoRR*, vol. abs/1707.01202, 2017. [Online]. Available: http://arxiv.org/abs/1707.01202

[7] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980