Deep Neural Networks for Object Enumeration:
Counting Items in Amazon Bin Images

Zihao Xu
Pomona College
zihao.xu@pomona.edu

Mariam Salloum
UC Riverside
msalloum@cs.ucr.edu

ABSTRACT

Estimating object count from images is a difficult problem that has a wide range of applications in medical research [1], crowd counting [2] and framing [3]. In this work, we examine the object counting problem for the Amazon Bin Images Dataset (ABID), which depicts bins of goods in an operating Amazon Fulfillment Center captured before shipment. This task is riddled with many challenges, including low image quality, occlusion, non-uniform object shape, and inconsistencies in the data labels. This work presents an in-depth study of the object counting problem, exploring both classification and regression using a deep-learning approach. We present two solutions - end-to-end training on ResNet [4] and model stacking with LightGBM [5] and ResNet – that show promising results for this difficult task. Further, we present layer visualizations to understand why the models are effective.

INTRODUCTION

Object counting is a well-studied task in the field of computer vision. In this work, we will:

- **Predict -** number of objects within images
- **Explore -** 3 methods of problem formulation (*Figure. 3*)
- **Employ -** deep learning and model stacking
- **Visualize -** activation maps of deep layers

POTENTIAL CHALLENGES

- **Integer Prediction -** not directly modeled by CNN
- **Variety -** many shapes, sizes, and colors of objects
- **Partial information -** occlusion by other objects/tapes, images cut-off or blurry
- **Bundle Deals -** packages or bundles cause confusion
- **Mislabeling -** wrong labels might be assigned to images

DATASET

Images from the **Amazon Bin Image Dataset (ABID)** depict bins of goods in an operating Amazon Fulfillment Center captured before shipment. The dataset also includes metadata that provides the number of items depicted in each image, along with a description of the item. Note, no localization information for each individual item is provided.

- **Fig. 1:** shows sample images with corresponding labels
- **Table 1:** provides some summary statistics of the dataset
- **Fig. 2:** shows the distribution of the object counts for all images. Note, given that the images with 0 – 5 items are the most common in the dataset, we focus on solving the object counting problem on this subset of images.

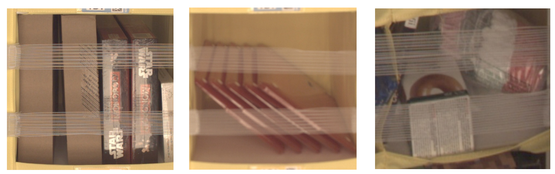

Figure 1 Sample images with 5 objects

Statistics	*Count*
Total Images	*536,432*
Images with less than 5 items	*361,967*
Max Item Count	*209*
Min Item Count	*0*
Average Item Count	*5.1*

Table 1. Summary Statistics

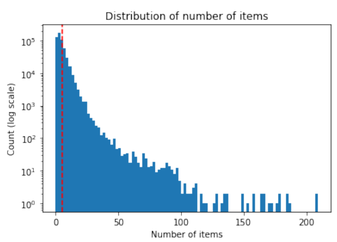

Figure 2. Distribution of Object Counts

APPROACH

Pre-process input images - resize to 224 x 224 and standardize

↓

Set up training (80%), validation (10%), and testing sets (10%)

↓

Optimizer - Adam, learning rate - 5e-3, batch size - 32, epochs - 100

↓

ResNet50 + data augmentation (*hvflip, rotation*)

↓

Test-time augmentation (*hvflips*) + layer visualization

↓

Extracting deep features at the last layer

↓

LightGBM using deep features and image sharpness

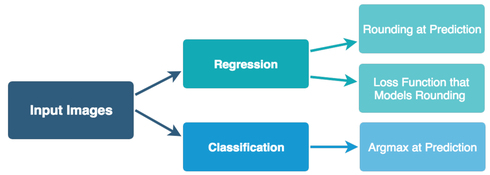

Figure 3 Possible Problem Formulation

EXPERIMENTAL RESULTS

As shown in *Fig. 3*, we formulated 3 different methods to tackle this integer prediction problem, which cannot be directly modeled by CNN. In particular, we explored:

- **Regression**: rounding at prediction time
- **Regression**: a loss function (*logcosh*) having similar effects as rounding
- **Classification**: argmax operation at prediction time

In addition to problem formulation, we also experimented with:

- **Data Augmentation**: horizontal/vertical flip, random rotation
- **Test-time Augmentation**: argmax on sum of probabilities for augmented test images
- **Model Stacking**: LightGBM with deep-features and image sharpness

Table 3 shows the model comparisons. *Fig. 4 plots* the final model architecture for *resnet_clf_GBM_sharp_TA*, our final model. *Fig. 5* illustrates its training process while *Fig.6* plots the test-time confusion matrix. The final test accuracy is 55.17%.

Model Name	*resnet_reg*	*resnet_reg_logcosh*	*resnet_clf*	*resnet_clf_TA*	*resnet_clf_GBM_sharp*	*resnet_clf_GBM_sharp_TA*
Acc.	48.21%	49.99%	53.38%	54.64%	53.89%	55.17%
MSE	0.7972	0. 8031	0.9441	0.8982	0.9003	0.8591

Table 3. Model Performances – Test Accuracy and Mean Squared Error

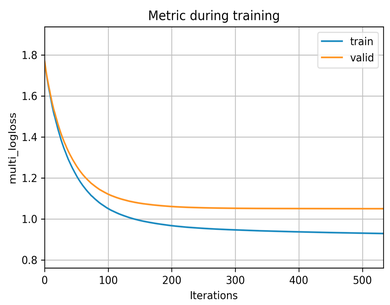

Figure 5. Training Process of resnet_clf_GBM_sharp_TA

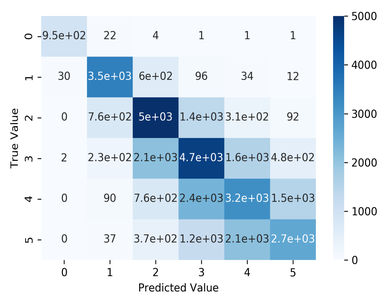

Figure 6. Test-time Confusion Matrix

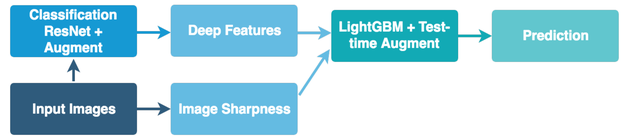

Figure 4. Final Model Architecture

LAYER & ACTIVATION VISUALIZATIONS

To understand how the model works, we plot

- **Fig.7**: activation maximization for filters within 3 layers from the trained ResNet50 model. This figure shows that the model is trying to capture shapes and textures of the objects within the images.
- **Fig. 8**: correlation between activation values in the 2 most predictive filters (LightGBM feature importance) and image labels. This figures show us that filters within the model are trying to capture ”typical scenes” of bin images with a specific number of objects.

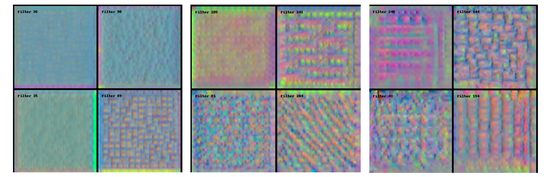

Figure 7. Filters from Layers *res3a_branch2b, res4a_branch2b, and res5a_branch2b*

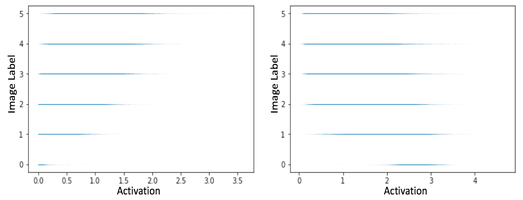

Figure 8. Activation Values against Image Labels

CONCLUSIONS

What we have accomplished:

- Explored the different methods of integer prediction using CNN
- Built a stacked classification model achieving 55.17% test accuracy
- Visualized filters and activation values from the trained ResNet model

Further directions:

- Explore other problem formulations such as ordinal regression
- Design a CNN architecture more suited for the object counting task
- Expand to predicting images with more than 5 objects

REFERENCES

[1] V. Lempitsky and A. Zisserman, “Learning to count objects in images,” in Advances in Neural Information Processing Systems , 2010.

[2] 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.

[3] M. Rahnemoonfar and C. Sheppard, “Deep count: Fruit counting based on deep simulated learning,” Sensors , vol. 17, no. 4, 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[5] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” in Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 3146–3154.