Amazon Inventory Reconciliation Using AI

Pablo Rodriguez Bertorello, Nutchapol Dendumrongsup, Sravan Sripada
Department of Computer Science, Stanford University, USA

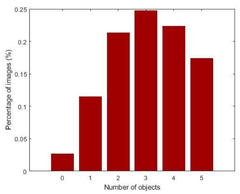
Motivation

Amazon Fulfillment Centers are bustling hubs of innovation that allow Amazon to deliver millions of products to over 100 countries worldwide with the help of robotic and computer vision technologies. Occasionally, items are misplaced while being handled, resulting in a mismatch between recorded bin inventory and contents of some bin images. The project consists of using a bin image dataset to count the number of items in each bin, to detect variance from recorded inventory.

Dataset

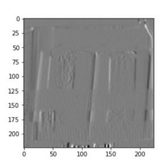
Amazon has made public the Bin Image Dataset. It contains images and metadata from bins of a pod in an operating Amazon Fulfillment Center. The bin images in this dataset are captured as robot units carry pods as part of normal operations. Bin Image dataset provides the metadata for each image from where number of items in bin can be derived.

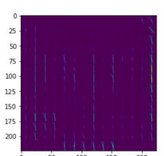
- Consists of 150,000 Images of different sizes
- Restricted project to only images with fewer than 6 bin items
- Histogram of distribution mass, for the number of object in a bin:
- Some images are occluded
- Even for a human, some images are difficult to classify



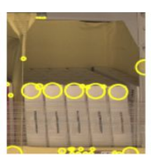
Preprocessing

- Resized all the images to 224 x 224
- Normalized images for zero mean, unit variance
- Split into 70% training, 20% validation and 10% testing
- Feature extraction:
  - Raw pixel: with and without extracted principal component analysis (PCA)
  - Histogram of gradients: with and without edge detection to remove the impact of the tape
  - Laplacian of gaussian
  - Horizontal flip Data Augmentation


Sobel – X Gradient

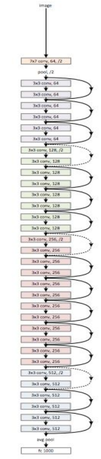

Histogram of Gradients


Laplacian of Gaussian


Laplacian of Gaussian

Algorithms

Convolutional Neural Networks (CNN)



Architectures : ResNet 34, ResNet 18

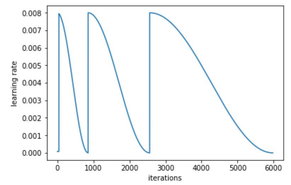
Cross Entropy Loss Function

$$\sum_{c=1}^{M} - y_{o,c} \log(p_{o,c})$$

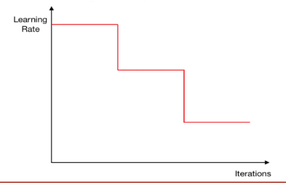
Softmax – Fully connected Layer

$$p(y = i|x; \theta) = \phi_i = \frac{\exp(\theta_i{}^T x)}{\sum_{j=1}^{k} \exp(\theta_j{}^T x)}$$

Stochastic Gradient Descent with Restarts – Cosine Annealing (SGDR)



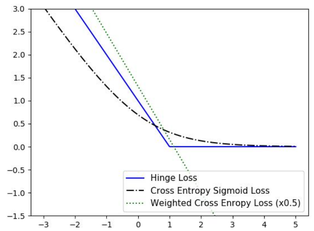
Stochastic Gradient Descent – Step Annealing (SGD)



Support Vector Machines

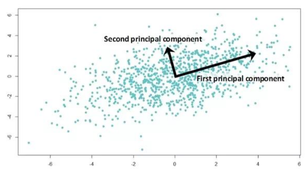
Hinge Loss

$$L_i = \sum_{j \neq y_i} \max(0, w_j^T x_i - w_{y_i}^T x_i + \Delta)$$



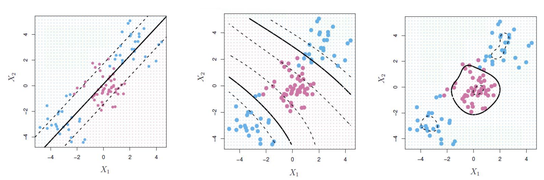
Dimensionality reduction using Principal Component Analysis

$$\mathbf{w}_{(1)} = \arg\max_{\|\mathbf{w}\|=1} \{\|\mathbf{X}\mathbf{w}\|^2\}$$



Kernels

Linear : $< x, x' >$
Polynomial : $(\gamma < x, x' > + r)^d$
Rbf: $\exp(-\gamma \left|\left|x - x'\right|\right|^2).\gamma$

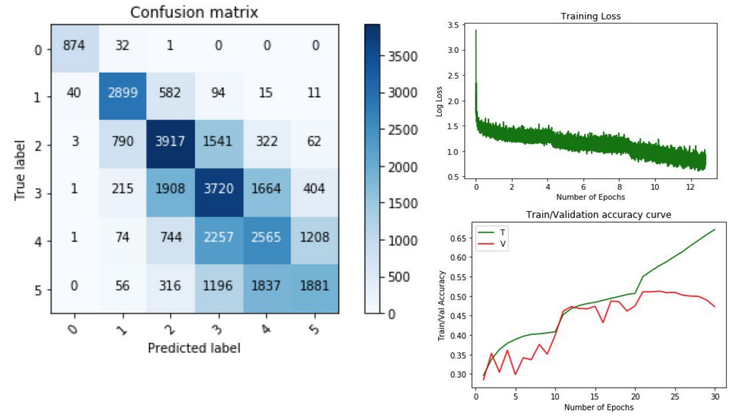


References

- Justin Johnson, Andrej Karpathy. Stanford CS231n: “Convolutional Neural Networks for Visual Recognition”
- Patrick H Winston. MIT 6.034: “Support Vector Machines”

Results: CNN

Model	Train Accuracy	Test Accuracy	Test Root Mean Square Error
ResNet 18 (SGD)	55.9	50.4	0.98
ResNet 34 (SGD)	55.2	51.2	0.99
ResNet 34 (SGDR)	57.8	53.8	0.94
ResNet 34 (Adam)*	62.3	56.2	0.90



Results: SVM

Bin Image Count	Blob Features	HOG	Raw pixel with PCA
0	0.038	0.33	0.71
1	0.17	0.12	0.24
2	0.22	0.29	0.32
3	0.53	0.26	0.48
4	0.22	0.29	0.28
5	0.018	0.24	0.12
Overall	0.26	0.26	0.32

Discussion

- We were able to improve the model accuracy by 75% using CNN over SVM, for an overall 324% over random
- Our best model is achieving over 97% accuracy on the images with no items and the accuracy goes down as the number of items in images go up

Future Work

- Remove the images from the training set where items are occluded
- Train the model on different training/val/ sets to reduce data distribution differences
- Evaluate ensemble methods to improve the accuracy
- Use class weights to assign higher loss to classes with suboptimal performance
- Use all 324K images for training instead of a sample of 150K images