Deep Learning in Crown Defects Inspection

James H. Chen
jchen1ca@stanford.ca

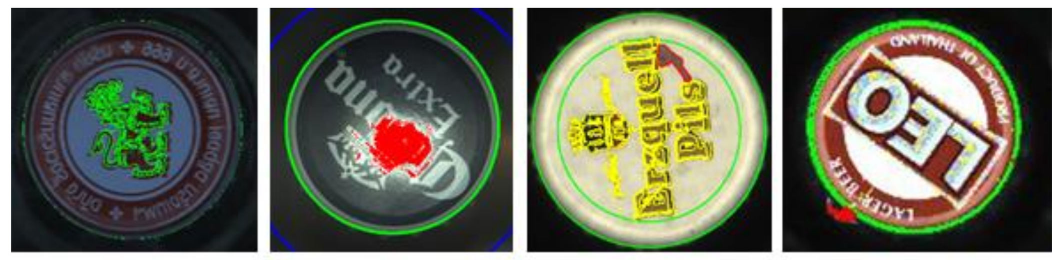
Video: https://youtu.be/mrpEOSEEJjQ

Motivation

In production line, such as beer packaging industry, majority products are the same, except a little amount of products with unacceptable difference, which we call defects, needing to be rejected from the conveyor.

Traditional machine vision defects inspection algorithms are customized to focus on user defined ROIs with specifically designed filters and thresholds, which requires specialists to configure it properly. For example, to detect various beer bottle crown defects, we need to use pattern matching to compare crown with different text, use special light and grayscale thresholding to detect dents, use regional grayscale checking analysis to detect crimp flare. But when customer asks to detect scratches, none of these tools can do the job

Can we use a single deep learning model to classify all the defects at once?



(a) Pattern Matching (b) Dent Inspection (c) Scratch Inspection (d) Flare Inspection

Datasets

To maximize the inspection capability, we collect total six classes in the dataset. When the container comes through the inspection system, each crown is classified as good crown, and defective crowns as listed in the table. The good crown will pass, while the defective classes will be categorized and rejected. Since product line downtime is very expensive, we keep the datasets to as minimum as possible, so the training time won't be too long. We use only 4 sample crowns for each class. This is to test how small amount of samples is needed to train a model smart enough to classify all the possible defects in that class. In order to reduce the run time, and increase the accuracy, the final images in dataset are cropped from 1920x1200x3 full size images and resizes to 128x128x3 resolution images and normalized by 255.

Good Crown y = 0 | Color Difference y = 1 | Text Difference y = 2 | Date Code y = 3 | Flare Defects y = 4 | Scratch Defects y = 5

Classes	Total Sample Crowns	Total Images	Training Set	Test Set
Good Crown	4	500	400	100
Color Difference	4	500	400	100
Text Difference	4	500	400	100
Date Code On	4	500	400	100
Crown with Scratch	4	500	400	100
Crown with Flare	4	500	400	100
TOTAL	24	3000	2400	600

Models

We use two deep learning models to evaluate the training speed, accuracy, and prediction run time.

1. Deep neural network with hidden layers: including 3 linear layers , 2 ReLu layers and finally 1 softmax function. The model structure:
   LINEAR -> RELU -> LINEAR -> RELU -> LINEAR -> SOFTMAX
   Optimized hyper parameters: Learning rate = 0.0001; minibatch=32; epochs = 40; Initialization = "Xavier"; Optimization = "Adam".
2. CNN model: constructed with ConvNet, pool, padding, flatten, FC filters layers, and ReLu and softmax activation functions. The model structure is:
   CONV -> RELU -> MAX POOL -> CONV -> RELU -> MAX POOL -> FLATTEN -> FC -> SOFTMAX
   Optimized hyper parameters: Learning rate = 0.002; minibatch=32; epochs = 40; Initialization = "Xavier"; Optimization = "Adam".

Use TensorFlow function to calculate the categorical cross-entropy as the cost. Use Adam optimization for both models. Use Xavier initialization which is more efficient than He_norm.

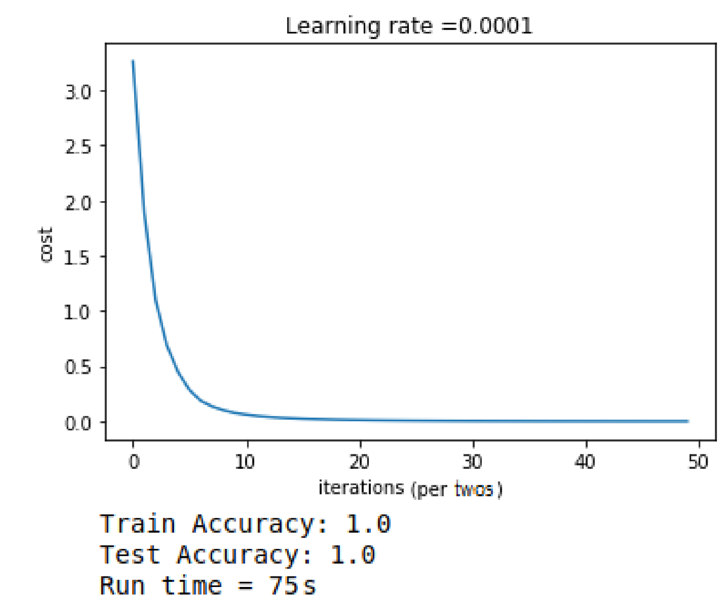
Hardware & Platforms

We use standard industrial vision engine built with 12 cores Intel Core i7-6700k, 16GB DDR4, in Ubuntu16.04, with no GPU used.

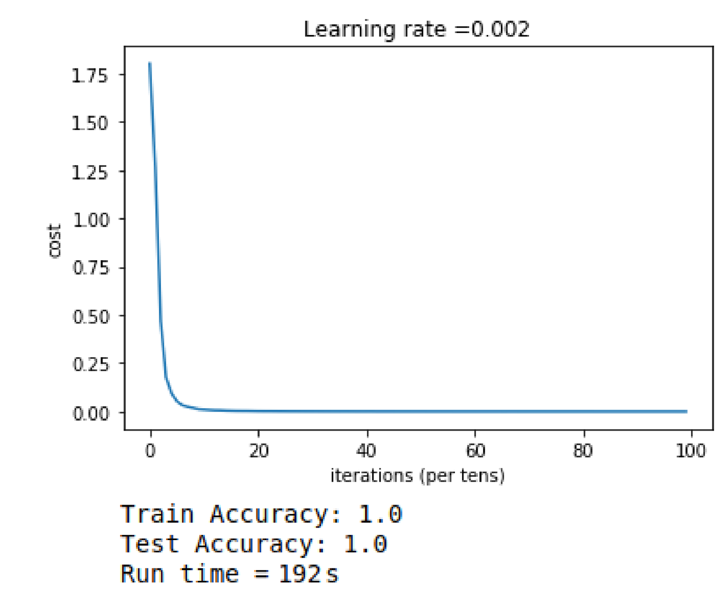
Items	Descriptions
MB	GA-Z170X-UD5
Processor	Skylake 2x Intel Core i7-6700K CP CPU@4.00GHz (No GPU)
Memory	2x 8GB DIMM DDR4
Caches	128KB x2 L0/L1, 1MB L2, 8MB L3
OS	Ubuntu 16.04
Platforms	Python 3.6.9, TensorFlow 1.6, vc2.0, Halcon 19

Results

During training, both models can quickly reduce cost to 0.000001 level after running 40 epochs. Both models can get test accuracy to 100%.



Model1: Neural Network Model



Model2: CNN Model

Other test results:

- The designed dataset is large enough to train for good models
- The features are simple and obvious. No need high epoch.
- Training time only need less than 4 minutes.
- Prediction run time is around 33ms.
- 120x120 image resolution is sufficient.
- 4 samples for each class is OK except scratch and flare classes.
- Features can be generated by augmentation, i.e. image rotation.
- In the six classes, color, text and date code difference is very easy to get trained properly with small amount of images. Scratch class needs more samples to get better performance. Crimp flare class is the hardest class which needs to feed more images.

Conclusion

Both models can get 100.0% accuracy for training, test, and validation after 40 epochs. Dataset can be as small as 40 training and 20 test images per class. The prediction run time is 33ms with 128x128 image resolution in Intel i7 CPU. A single deep learning tool can do inspections that used to use 6 independent tools.

Future works:

(1) The smallest scratch or flare defects can be inspected.
(2) Deploy the application to production line test.
(3) Balance precision, recall and accuracy, to reduce false negative.
(4) Train the same models for product label tear, flag, wrinkle, peel, faded color, and floater inspection.