
Deep Learning in Crown Defects Inspection

James H. Chen

Stanford University
jchen1ca@stanford.edu

Abstract

Can Deep Learning (DL) replace traditional machine vision algorithm? This project answers the question by applying deep neural network model and CNN model to inspect bottle crown defects. Both models can accurately classify good crowns from defective crowns such as wrong color crown, wrong text crown, date coded crown, scratched or dented crown, and even the crown with crimp flare. Further experiments demonstrate that deep learning algorithm can run in industrial CPU processor. It can perform the same speed as traditional machine vision algorithm, and can reach 1200 containers per minutes. The training data set can be limited within 500 images for each class. These images can be regenerated from a few delegate images by image augmentation such as rotation. The total model training time is less than 5 minutes. The run time in validation is 33ms. The models can detect new scratch and flare defects even it's in different location with different size and shape. It is becoming true that if human eyes can see it, then deep learning model can tell it[1].

1 Introduction

Traditional machine vision defects inspection algorithms are developed to focus on user defined ROIs with specifically designed filters, such as Gaussian or Sobel, and grayscale or gradient thresholds. It is custom solution for specific inspections in limited conditions. It requires specialists to configure the systems properly, and strategically balance the thresholding between false positive and false negative. When there is new kind of defects coming up, the current inspection tool may become invalid, therefor need to develop new tool. 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. [2]

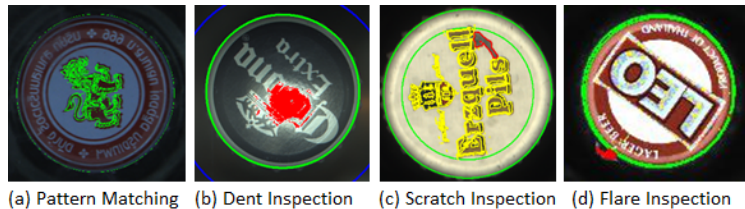


Figure 1: Traditional machine vision methods (Copyright of Filtec[2])

Deep learning model, on the other hand, can be trained very smart to find any difference from the good class, especially in the highly repeatable production line. For example, if the capper machine brings an unknown scratch defects in a period of time, then we can feed the model with those defective crowns, so the model can recognize it as a new category, and tell rejector to reject it. We don't need to develop any new inspection tool. No need to design a new light or mechanical system to highlight the scratch.

In this project, we collect all the possible top view inspection tasks from Filtec Intellect systems, and create datasets of six classes, which includes all the inspection features. We train and test with two different neural network models, deep

neural network model, and a CNN model. This images are cropped, resized and normalized. We then use these models to predict new images captured from different crowns. Both models can successfully predict the images correctly.

2 Related work

In the past decade, there are many articles for deep learning applications in surface defects inspection [3], multi-type texture defects inspection [4], tunnel crick defects inspection[5], PCB inspection [6], deformable patterned fabric defect detection[7], and articles talking about build cognitive defects inspection tools to industry[8].

There is no article on how to classify images with very similiar background. And there is very limited number of articles discussion how to simplify deep learning models to fit industrial need, such as training time and validation run time requirement.

Investigate the accuracy and precision, Filtec product owner claims that, for severe defects, such as glass shard sinker, or high pressure danger bottle, the minimum detection rate (false positive) needs to be 99.9%. The false rejects (false negative) needs to be lower than 0.1%. For other unimportant defects inspection, the minimum detection rate can be 80%. The false rejects still needs to be lower than 0.1%.

Eliminate background noise can improve the model performance. This is done by applying a bounding box to crop the image to just a little bigger than the target, then rescale image to fixed size [9].

3 Dataset and Features

To maximize the inspection capability, there are total six classes in the dataset (Table 1). When the container comes through the inspection system, each crown is classified as good crown, crown in different color, crown with different text, crown with or without date code, crown with scratch defects, crown with crimp flare defects. The good crown will pass, while the other classes will be categorized and may be rejected.

To reduce the training time and simulate the real field application, we use small dataset with limited amount of crown samples. We prepare 4 different crowns for each class, total 24 sample crowns. Then crimp them to 24 containers, and put these containers on the conveyor. Run the conveyor for 125 cycles. Filtec Intellect inspection machine captures total 3000 raw images and save to SSD.

Classes	Total Sample Bottles	Total Images	Full Set Train	Full Set Test	Partial Set Train	Partial Set Test
Good Crown	4	500	400	100	20	10
Color Difference	4	500	400	100	20	10
Text Difference	4	500	400	100	20	10
Date Code On	4	500	400	100	20	10
Flare Defects	4	500	400	100	20	10
Scratch Defects	4	500	400	100	20	10
Total	24	3000	2400	600	120	60

Table 1: Datasets and components

The images were grabbed from Filtec Intellect machine in February. The raw image resolution is 1920x1200x3.



(a) Filtec Intellect Image Acquisition Head; (b) Full Size Image (1920x1200)

Figure 2: Source Image Acquisition

In order to reduce the run time, We use Halcon procedure to allocate crown center position and size, then crop around the crown and resize it to 128x128x3 resolution (and 240x240x3 which is the max size for Halcon deep learning library). Image grayscale is normalized by 255.

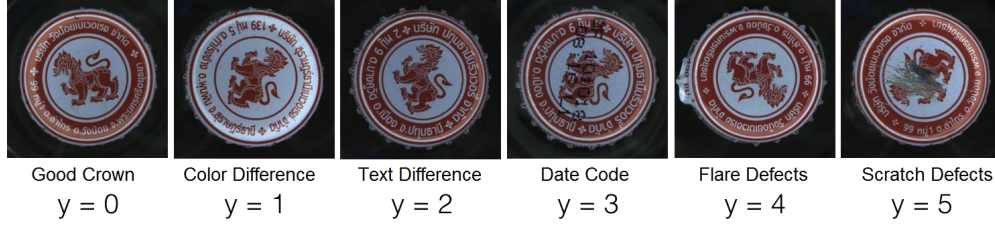


Figure 3: Classes of Crown Defects

4 Methods

This is multi-class image classification application, which is typical defects inspection model in computer vision[10]. We test with two different models to evaluate the speed and accuracy performance. Both models are based on Stanford CS230 Deep Learning 2020 Winter course models.

4.1 Deep Neural Network Model

This model uses pixels as input features, and build neural network with hidden layers including linear function and activation function. The activation function we used is ReLu function which does not activate all the neurons at the same time. Since the cropped images are very similar, and the difference is obvious, we use 3 linear layers, 2 ReLu layers and 1 softmax. We use softmax to replace sigmoid since there is multi-class output. The mode structure is:

LINEAR -> RELU -> LINEAR -> RELU -> LINEAR -> SOFTMAX.

TensorFlow sigmoid cross entropy with logits loss function is used to calculate the categorical cross-entropy as the represented below:

$$J = -\frac{1}{m} \sum_{i=1}^m (y^{(i)} \log \sigma(z^{[2](i)}) + (1 - y^{(i)}) \log(1 - \sigma(z^{[2](i)})))$$

4.2 CNN Model

CNN model is constructed with ConvNet, pool, padding, flatten, FC filters, and ReLu, softmax activation functions. Multi-layer deep CNN model can recognize complex image features. This crown defects inspection is a typical project of CNN based deep learning in computer vision applications. It uses labeled samples with supervised learning. Multiple layers of deep learning algorithm are used to increase the accuracy. All three image channels are included in the input. Final result can classify the defects features. The procedure is:

CONV -> RELU -> MAX POOL -> CONV -> RELU -> MAX POOL -> FLATTEN -> FC -> SOFTMAX

4.3 Hardware and Platforms

We use currently existed Filtec Intellect vision engine built with 12 cores Intel Core i7-6700k, 16GB DDR4, in Ubuntu 16.04, with no GPU support. We get very short validation run time, as explained in 5.4.

Systems used in the project:

Items	Descriptions
MB	GA-Z170X-UD5
Processor	Skylake 2x Intel Core i7-6700K CP CPU@4.00GHz (12cores, 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 12, Jupyter Notebook 6.0.3

Table 2: Systems used for the test

4.4 Others

Adaptation test: Test with newly create new defects with size and shape different from the trained sample bottles. Test with images with different brightness, though there is image grayscale rescaling Test with images grabbed from changed

mechanical and optical configuration, such as different conveyor speed, different guide-rail width (created images with parallax influence) Apply the application to other bottles such as PET bottle.

Test initialization methods, both Xavier and He initialization methods are used in the test to evaluate speed and gradient decay.

5 Experiments/Results/Discussion

The target of this project is to use the smallest dataset to train a fast model with limited amount of hyper parameter settings. Except optimizing the hyper parameters, there is a sequence of engineering test including dataset size test, image resolution and run time test, minimum number of samples test, image augmentation test.

5.1 Hyper Parameter Tuning and Optimization

Three main hyper parameters, learning rate, minibatch size and epoch number, are trained[11]. The final choice is, for deep neural network model, the learning rate is 0.0001, the minibatch size is 32, the epoch number can be as low as 20, but recommended to use 40. For CNN model, the learning rate is 0.0002, the minibatch size is 32, the epoch number is 40. There is no obvious difference when change ConvNet filter size. The reason maybe because the image resolution is high enough. Comparing with these two models, deep neural network model is relatively faster with the same accuracy. The gradient decay is consistent as the cost function curve is very smooth. This is benefited from the fact that images are cropped to the the same size with very similar background. When the models are trained more than 40 epochs, there is no performance difference between them except CNN model is slower. When run only 20 epochs, we can see some difference. There are other tests. Adam optimization is the best optimization method. Xavier initialization is better than He-norm initialization.

5.2 Dataset Size Test

This is to test what is the minimum amount of training images needed for a good model training. First, run full dataset of images (training set 2400, test set 600), with 240 x 240 resolution images. See below results. Both models can very easily reach the optimal point with lowest cost in 20 epochs. Both models can get training and test accuracy to 1.0.

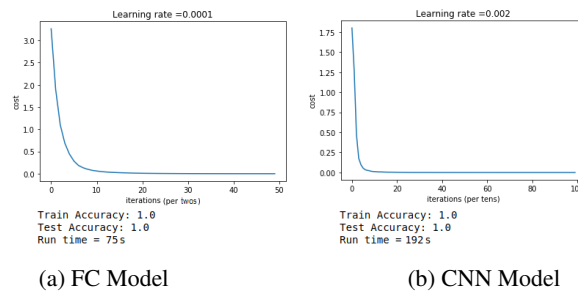


Figure 4: Full size of data sets training (400 training images, 100 test images per class)

Then run partial dataset test. Use 5% of the full images, i.e. use 20 images for training and use 10 images for test for each class), we can get training accuracy to 100.0%, and test result to around 85% (Figure 5). Finally add training set to 40 images, and keep test set as 10 images. We get both training and test accuracy to 100.0%

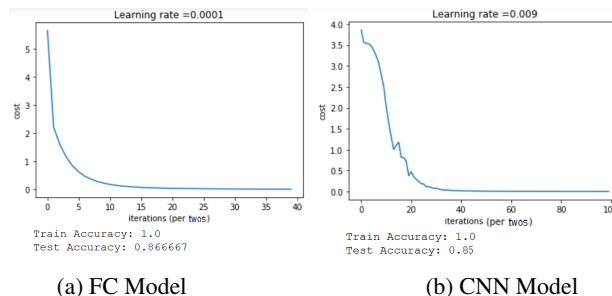


Figure 5: Partial data sets training (20 training images, 10 test images per class)

5.3 Image Resolution and Run Time Test

The training time and validation run time is very short. For deep neural network model, with 3000 images at 128x128 resolution, the training time is 16s when run 20 epochs, and validation run time is between 33ms to 39ms. At 240x240 resolution, the training time is 198s when run 100 epochs, and validation run time is around 73ms.

5.4 Minimum Number of Samples Test

What is the minimum number of sample crowns needed to train a model that can classify all kinds of defects with different size, shape and orientation? Below we test with two pairs of defects. We train the model with image (a) and image (c), then evaluated it with image (b), including flare defects with different orientation and size, and image (d), including scratch defects with different position. Both tests get the good results. Model trained with image (a) and (c) can be used to correctly classify image (b) and image (d) as flare defects and scratch defects specifically.



Figure 6: Model can predict well even validation crown is different from training crown

5.5 Image Augmentation Test

This is a very interesting test. We use a single image from each of the classes, and rotate the image 360° to get 500 images for that class. Then we use these 500x6 images to train the model. Surprisingly, the model can get 100.0% accuracy for the small dataset, and can successfully validate all the other samples. This is a big benefit for the final users to save the training time by simply saving one image for each sample crown. Then the software can rotate and generate a dataset of images for the model training.

6 Conclusion/Future Work

We tested two different models. Both models can get 100.0% accuracy for training, test, and validation after 40 epochs. It is doable to use small amount of sample images (40 training and 20 test images) to get a good result. Using rotation to create image augmentation is a time saving method to to set up dataset. The validation run time is very short. With image resolution as 128x128 pixels, the run time is 33ms in Intel i7 CPU, which can run more than 1200 containers per minute. We can use deep learning algorithm to upgrade the current machine vision algorithms in Filtec.

Future work will include the following: (1) How small scratch or flare defects can be inspected. (2) Deploy the application to Filtec Intellect vision engine and test online. Need to consider details such as how to design hyper parameters in user interface, how to adjust deep learning layers, how to augment images to save down time. The current hardware is powerful enough to run deep learning algorithm. (3) How to balance precision, recall and accuracy, to reduce false negative. (4) Encouraged by this success, we will investigate the possibility to apply deep learning to other applications such as beer bottle label tear, flag, wrinkle, peel, faded color, and floater inspection.

In general, we are very pleased to see that deep learning method is our next generation of technology. We will create a few generic tools for our field engineers. So they can easily train the models based on their immediate need to satisfy customers new requirements which is not included in our inspection toolbox.

7 Contributions

This is a solo project.

Code: <https://github.com/jchen1ca/macadamias/projects/1>

Video: <https://youtu.be/mrpE0SEEJjQ>

References

- [1] Thomas Dean, Mark A Ruzon, Mark Segal, Jonathon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1814–1821, 2013.
- [2] Shangyong Tang, Guilan Wang, Haiou Zhang, and Rui Wang. An online surface defects detection system for awam based on deep learning. In *Solid Freeform Fabrication 2017: Proc. of the 28th Annual International Solid Freeform Fabrication Symposium—An Additive Manufacturing Conference*, volume 81, 1965.
- [3] Ruoxu Ren, Terence Hung, and Kay Chen Tan. A generic deep-learning-based approach for automated surface inspection. *IEEE transactions on cybernetics*, 48(3):929–940, 2017.
- [4] Young-Jin Cha, Wooram Choi, Gahyun Suh, Sadegh Mahmoudkhani, and Oral Büyüköztürk. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, 33(9):731–747, 2018.
- [5] Konstantinos Makantasis, Eftychios Protopapadakis, Anastasios Doulamis, Nikolaos Doulamis, and Constantinos Loupos. Deep convolutional neural networks for efficient vision based tunnel inspection. In *2015 IEEE international conference on intelligent computer communication and processing (ICCP)*, pages 335–342. IEEE, 2015.
- [6] Yu-Shan Deng, An-Chun Luo, and Min-Ji Dai. Building an automatic defect verification system using deep neural network for pcb defect classification. In *2018 4th International Conference on Frontiers of Signal Processing (ICFSP)*, pages 145–149. IEEE, 2018.
- [7] Yundong Li, Weigang Zhao, and Jiahao Pan. Deformable patterned fabric defect detection with fisher criterion-based deep learning. *IEEE Transactions on Automation Science and Engineering*, 14(2):1256–1264, 2016.
- [8] Adnan Masood and Adnan Hashmi. Cognitive computing recipes.
- [9] Chang Liu, Yu Cao, Yan Luo, Guanling Chen, Vinod Vokkarane, and Yunsheng Ma. Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment. In *International Conference on Smart Homes and Health Telematics*, pages 37–48. Springer, 2016.
- [10] Francois Chollet. *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG, 2018.
- [11] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.