Deep Learning Approach to Planogram Compliance in Retail Stores

Timothy Chong, Idawati Bustan and Mervyn Wee

Machine Learning, Computer Science
Stanford University, CA
{timchong, idawati, wyrm}@stanford.edu

Abstract – In this paper, we will examine different types of training images to determine which is the most suitable in achieving image classification of products on the shelves of retail stores. Being able to classify these products will augment inventory management in retail store that is being performed by Navii - a robot that was developed by Fellow Robots, Inc. A mixture of images collected by Navii and images from the Internet were collected, split into 3 types of training sets, and used to create 3 CNN models. Comparing the accuracy of these models over the same test set will lead us to the choosing the most suitable type of training data.

I. INTRODUCTION

Fellow Robot's latest robot model, Navii, is able to support inventory management in stores. Equipped with 3 cameras, it will drive around the store and take pictures of items on the shelves. These pictures will then be analyzed using machine learning models to provide valuable information, which can help with tasks such as planogram compliance.

A planogram is a model that specifies exactly how products should be displayed on the shelves in order to ensure maximum sales, and planogram compliance ensures that the products on display are in accordance with the planogram. There is a demand for planogram compliance in retail stores since it directly affects the stores' sales. However, planogram compliance is a time consuming and labour intensive task that retail companies are seeking to automate.

The objective of this paper is to recognize on-shelf products so that pictures scanned by Navii can be used for planogram compliance. Specifically, we would like to be able to detect misplaced items. For example, if a hammer was placed on a screwdriver shelf, we would like to be able to detect that the hammer is in the wrong shelf, and to notify the store to rectify the misplaced item. In order to detect whether an item is in the right shelf, there is a need to first identify that item. Once it is classified, it can then be compared to the planogram to see if the item matches the planogram.

In short, the Convolutional Neural Network (CNN) deep learning model will receive images of products on shelves in retail stores, and it will output a prediction on what product is in that image.

II. RELATED WORK

Automated Planogram Compliance

Traditional solutions to planogram compliance involve a store manager or employee walking around with a clipboard and verifying compliance. Higher tech solutions involve pictures taken with smartphone or iPad, then compared to the planograms. As presented in [6], this process is still labour-intensive and time-consuming with subject to error rates of up to 20% . Meanwhile, real-time product detection system requires high quality of training data [9]. Another study on tobacco packages [10] presented that satisfactory on-shelf product detection is possible even with low computational power using Support Vector Machine (SVM).

Convolutional Neural Network on Image Classification

In object classification, CNNs have become extremely popular due to their high success rates in accurately recognizing objects.

In pattern and image recognition applications, the best possible Correct Detection Rates (CDRs) have been achieved using CNNs. A study presented that CNNs have achieved a CDR of 99.77% using the MNIST database of handwritten digits [2]. Another study showed 97.47% CDR with the NORB dataset of 3D objects [3], and a CDR of 97.6% on ~5600 images of more than 10 objects [4]. CNNs not only give the best performance compared to other detection algorithms, they even outperform humans in cases such as classifying objects into fine-grained categories such as the particular breed of dog or species of bird [5].

III. PRELIMINARY DATASET

A retail store has thousands of products in its inventory. As a first step towards planogram compliance of the entire store, we focused on classifying eight different kinds of products. This will give us an idea of the effectiveness of our approach, and provide insight into developing an accurate model.

Fig. 1. Example of images of a class of product from the training set that shows slight variations between each of them.

Our datasets were divided into two parts, a training set and a test set. Since our objective was to recognize classes of product in a retail store, we decided to train and test the model using only pictures of products from an actual retail store. To make the model robust in identifying classes of products, we took pictures of these products at slightly different angles. The pictures taken by the robot during its daily rounds were fairly consistent, but had slight variations due to tolerances in the specified location and angle at which to take the photos.

Pre-processing of the product image dataset was done with TensorFlow image processing, which is included on the final layer of InceptionV3. Each image will be pooled into a 2048-dimensional vector, and then a softmax layer will be trained on top of this representation.

IV. ALGORITHMS

CNN is made up of a series of convolutional and pooling layers, in which in the final layer is fully connected at the output neurons with a softmax layer to give the confidence of the prediction (Fig. 2.).

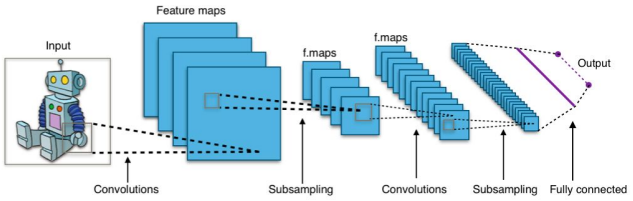

Fig. 2. General architecture of Convolutional Neural Network

A convolutional layer is basically a set of learnable filters (kernels) which represents a specific part of the image by preserving the spatial relationship between pixels. It is activated when it detects some specific type of feature at some spatial position of the input. The pooling layer (subsampling) reduces the dimensionality of each feature map but retains the most important information. This is done by a few common methods; max, average, sum pooling. The result is a smaller and more manageable feature dimension, with lesser number of parameters and computations.

The final fully connected layer involves a softmax function which will help us make the prediction, by exponentiating and then normalizing the inputs.

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

The output of the softmax function is used to represent the categorical distribution that gives us a list of values from 0 to 1 that add up to 1, which represent the probabilities (confidence) of each class prediction.

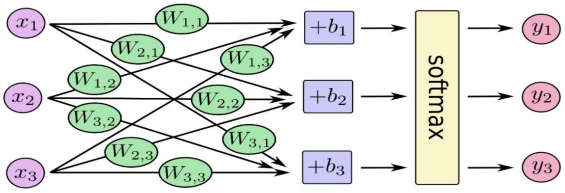

Fig. 3. Visualisation of softmax function implementation in the final layer of CNN models.

For this project, we used TensorFlow and did transfer learning on a pre-trained CNN Image Classifier, InceptionV3 which has been extensively trained on data from ImageNet. Transfer learning was the best option as we do not have the resources to train a CNN from the ground-up such as time, computing power and large training dataset. Throughout this process, we first came up with a preliminary approach, before improving upon it on the second iteration.

V. PRELIMINARY RESULT

We used a 10-fold cross-validation to determine our testing accuracy. For every epoch iteration, a random 90% of the dataset will be used for training while the remaining 10% are for testing. The predicted values are compared to the actual labels and weights are then adjusted through back-propagation. We tweaked the hyper parameters such as learning rate and training batch size to achieve higher quality estimates of the gradient within reasonable training time. This is done by plotting the loss function (cross entropy) and watching how it converges over each training iteration. Seen from the graph of Fig. 4., the model managed to reach 100% validation accuracy after about 100 iterations. We were able to achieve good predictions within by training and testing the model using images from one retail store.

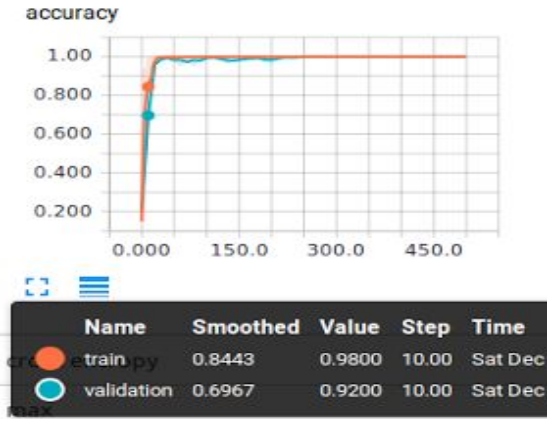

Fig. 4. Accuracy graph of training and validation steps for the CNN deep learning mode for first iteration.

However, despite obtaining good results, we realized that this would likely have been mostly due to the fact that the training and testing sets are vastly similar. In this particular case, overfitting the model gave us excellent predictions on the testing set. However, since Fellow Robots was trying to scale up to put inventory management robots in many retail stores, the model that was overfit on images from only one store may not be able to accurately classify images from other stores. Furthermore, if the retail stores were to stock up on a new product within any of the classes, the model would not be able to perform satisfactorily.

From a business and engineering standpoint, it made little sense to retrain the model every time a new retail store became a new client or whenever a new product was introduced into a store, so it was imperative that we improved on the model such that it would be able to make accurate predictions even on new products or in new retail stores.

VI. NEW METHODOLOGY

We recognized that the shortcoming in the preliminary approach was that we did not have enough images of different products within the eight targeted classes. The model was unable to generalize well because the training set was composed of multiple images of the same product from one retail store, even if there were slight variations between these images. To overcome this problem, we had to introduce more images of different products so that the model could extract the essential features of the targeted classes.

We decided that the best way of introducing many different types of products to model would be to download images of products from the Internet and add them to the training set. For each of these eight products, we collected images of as many different types of that product from a pool of at least 100 images.

Fig. 5. Training images of different classes of on-shelf products taken from the internet.

Fig. 6. Testing images of eight different classes of on-shelf products taken by NAVii - the robot from the home improvement store.

We hoped that adding images from the Internet would help solve the two aforementioned problems; namely, the cases where a new product was introduced, or if the robot was operating in a new retail store.

To evaluate the effectiveness of adding images from the Internet, we created three machine learning models: M1, M2 and M3. For M1, we trained it using images taken only from retail store X. M2 was trained using images solely taken from the Internet. M3 was trained using a mix of images from the Internet and retail store X.

All three models were tested on the same test set in order to provide a consistent basis of comparison. The test set consisted of 50 new images. 25 of these images were of product models that were included in the training set but taken in a different orientation, while the other 25 images were of new product models that were not used in the training set at all. Splitting the test set into seen and unseen products helped us to identify how well the models were able to predict products already in the store and new products that the store might stock up on in the future.

VII. RESULTS AND ANALYSIS

Model's result and accuracy on prediction

Our results showed that M3 performed the best with overall accuracy of 89.3%, tested with 50 test images of each classes of product. This was followed by M1 at 87.5%, and then M2 at 75%. (See Fig. 7.).

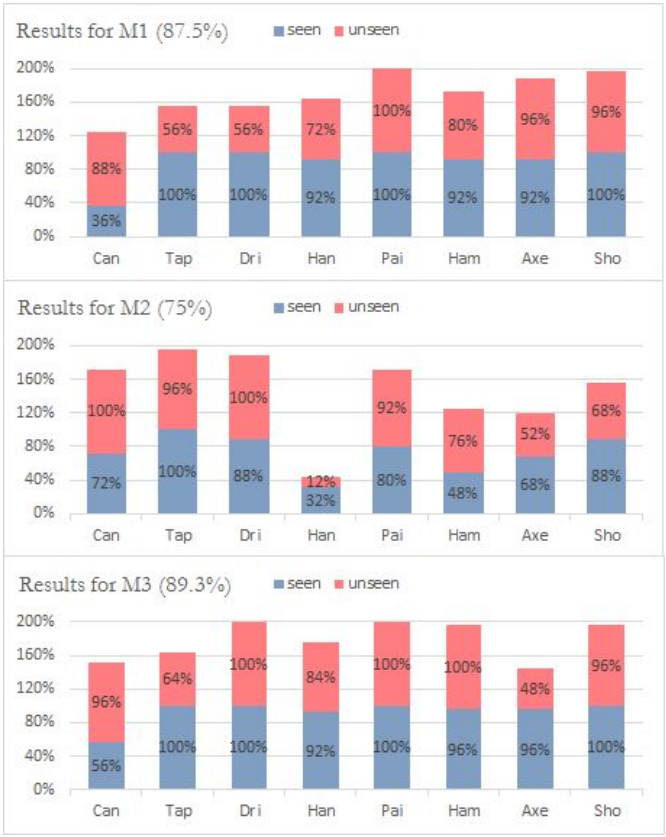

Fig. 7. Accuracy of results for three different models (M1, M2, M3) for two test sets.
{Can - Aerosol cans | Tap - Measuring Tape | Dri - Drills | Han - Hangers | Pai - Pail | Ham - Hammers | Axe - Axe | Sho - Shower Heads}

M3 consistently made better predictions than the other two models on almost all the eight classes when identifying both seen and unseen products.

This seems to indicate that the CNN makes better predictions when it has been trained on both store images and Internet images. Intuitively, we would expect M3 perform better on unseen products because the model was exposed to a larger variety of products, so it would be able to generalize better and avoid overfitting.

Misclassification of products analysis

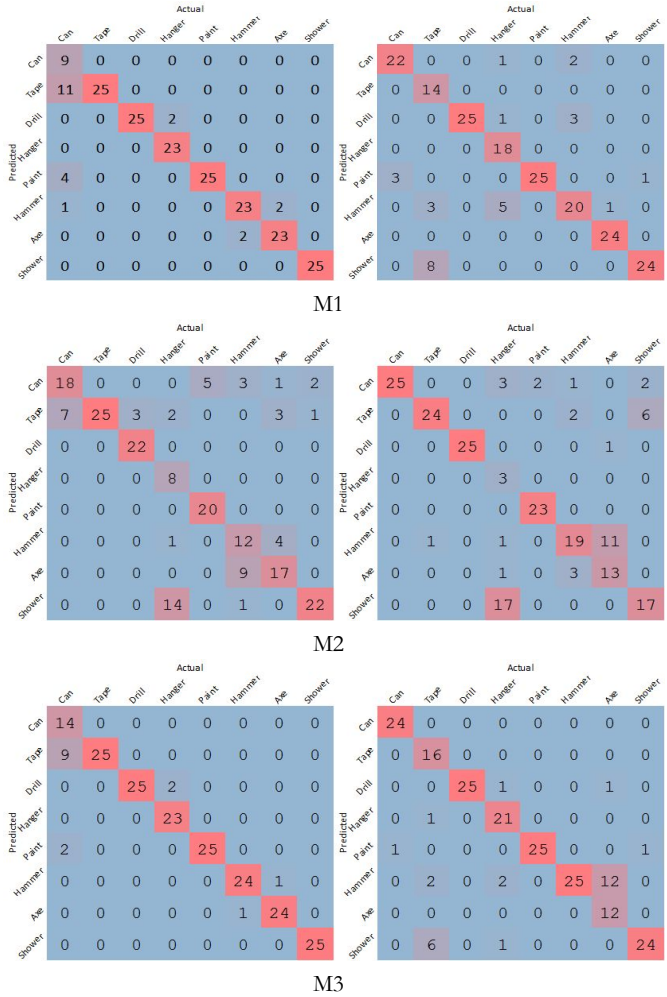

Fig. 8. Confusion Matrix of M1, M2 and M3 with left matrix for classification of existing products models and right matrix for unseen product models.

The confusion matrices (Fig. 8.) show detailed results of the testing and the products' misclassification pattern. According to the confusion matrix, M1 and M3 mostly classified the images well. However we observed consistent misclassification of aerosol cans as measuring tapes on seen products. Images in Fig. 10. were consistently classified as measuring tape across the three models with prediction confidence of lower than 40%.

Model/Avg PR	Precision	Recall
M1	0.896	0.875
M2	0.783	0.733
M3	0.919	0.893

Fig. 9. Summary of precision/recall values of each model.

The precision and recall score of a model is a more detailed representation of the robustness of the model. We derived the precision/recall values for each of the model from the confusion matrix above by calculating the true positives over the total predicted positives for precision, and true positives over number of positive cases. We took the average since it is a multi-class problem with two distinct testing sets. From the scores tabulated, we can see that M3 does a better job here in retrieving instances that are relevant, and thus is the best model overall.

Fig. 10. Images of aerosol cans consistently being misclassified as measuring tape with prediction confidence of 0.20 to 0.38

Analysis of impact of training images on prediction model

Our results suggest that our initial hypothesis that M1 was being overfitted was validated. M1 performed more poorly than M3 on unseen products, indicating that it did not actually generalize well on the product itself, but also took into account the background image and noise..

In the initial hypotheses, we also suspected adding images from the Internet would cause the model to suffer. However, the results prove otherwise for M3, as it performs better when predicting product that it had seen before.

It turned out that using images from the Internet was a double edged sword. M2 that is trained only using online images performs relatively bad as compared the other two. For some products like aerosol cans, measuring tapes, and hammers, augmenting the dataset with Internet images increased predictive performance. For other products like axes and shower heads, adding Internet images proved to be detrimental or have no effect. This is probably because online images looked different from on-shelf product images, especially on some products: clothes hanger, shower head and axe.

Case in point, shower heads and axes were often misclassified in M2. We found that the difference lay in whether the images from the Internet were actually similar to the products in the store. As previously seen, since the shower heads from the Internet were different from those in the store (Fig. 11.), there was no improvement in performance. For axes, as shown in Fig. 12., we found that the images from the Internet contained images that even a human would not classify as axes.

Fig. 11. Contrast between (top) images of training set from the Internet for M2 & M3 and (bottom) images of testing set from on-shelf products

Fig. 12. Inaccurate training images of axes taken from the Internet

It was of little wonder that M3 performed poorly on axes. For the rest of the products excluding shower heads, the images from the Internet were highly similar to the products in stores, thus explaining the good predictions made for these products.

The worst result comes from hanger of M2 that is heavily misclassified as shower, shown by the confusion matrix in Fig. 9. As previously discussed, the underlying reason for misclassification would be due to inaccurate training images from the Internet.

Confidence of prediction of M3

Another point worthy of consideration is how confident the model was in its predictions indicated by the softmax function output. Since M3 was the most effective model, in this part the discussion will only involve M3. Despite getting good predictions, further investigations revealed that it was possible for further improvements in raising the probability of a correct prediction.

For example, M3 scored 92% on seen hangers. Upon closer inspection, we found that 10 out of 23 correct predictions had a probability of between 40% to 60%. For unseen hangers, M3 scored 84%, and 15 out of 21 predictions had a probability of between 27% to 60%.

Fig. 13. Seen hanger models in the home improvement store X

Fig. 14. Unseen hanger models from other home improvement retail stores.

This means that although the model classified these images correctly, prediction confidence was low for most of them. Images in Fig. 13 and 14 show that for hangers, some hanger models or the way the hangers are displayed in the store is different from the images in the training set.

Fig. 15. Hangers from the training set of M2 and M3

In order get better predictions, we would need to reconstruct our training set with more varied images to better address the variations in displays in stores. Furthermore, even though prediction success rate is good, it is important that we analyse the underlying confidence to see if there is anything that is not performing so well.

VIII. FURTHER CONSIDERATIONS

Our work thus far has shown encouraging signs that it is possible for us to use images from the Internet combined with images from one store to make accurate predictions on images from other retail stores.

This would be useful from the engineering and business perspective because this suggests that there is no need to retrain the machine learning model each time a new retail store becomes a client, or when a new product is stocked in the store.

More work needs to be done to increase the number of classes of products, and to increase the number of stores that the test set images were taken from. Doing this will show to a greater extent the feasibility of this approach.

One more consideration is that the products we've used in this experiment are significantly different from each other in terms of shape. With the exception of axes and hammers, the rest of the products are easy to distinguish from each other. Once we increase the number of classes of products, we might face the problem where the model confuses one product for another. Indeed, in M3, 12 out of 25 of unseen axes were misclassified as hammers. This might be a problem that we would encounter in the future.

On a technical side, besides just collecting more data, we can tweak the hyperparameters to the model, which may help achieve optimal results. Areas to explore in this aspect are convolutional filter width, pooling filter width and the dropout strategy in neural networks.

To summarize, focusing on a small number of classes of products have shown promising results that we could further develop by extending the scope of the project. More work needs to be done to give greater confidence that our approach can be done on a larger scale.

IX. ACKNOWLEDGEMENTS

We would like to thank our TA Rishabh for his guidance, and Professors Andrew Ng and John Duchi for the great course. We would also like to thank our colleague at Fellow Robots, Sivapriya Kaza, for her advice and mentorship. It has been a wonderful and tiring journey thus far, and we hope that that our first machine learning project will be a stepping stone to greater endeavours.

X. REFERENCES

[1] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *IJCAI'11 Proceedings of the Twenty-Second International Joint conference on Artificial Intelligence,* volume 2, pages 1237-1242, 2011. doi:10.5591/978-1-57735-516-8/IJCAI11-210D.

[2] C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR),* 2015. arXiv:1202.2745

[3] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. In *Neural Computation*, volume 18, 7:1527-1554, 2006. doi: 10.1162/neco.2006.18.7.1527

[4] S. Lawrence, C. L. Giles, Ah Chung Tsoi, and A.D. Back. Face recognition: a convolutional neural-network approach. In *IEEE Transactions on Neural Networks,* volume 8 1:98-113, 1997. doi: 10.1109/72.554195

[5] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning,* pages 609-616, 2009. doi: 10.1145/1553374.1553453

[6] S. Liu and H. Tian. Planogram compliance checking using recurring patterns. In *ISM*, 2015.

[7] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *International Conference on Machine Learning*, 2007.

[8] O. Russakovsky, et al. ImageNet Large Scale Visual Recognition Challenge. In *Computer Vision and Pattern Recognition (CVPR),* 2014. arXiv:1409.0575

[9] T. Winlock, E. Christiansen, and S. Belongie. Toward real-time grocery detection for visually impaired. In *CVPRW,*pages 49-56, 2010.

[10] G. Varol and R. S. Kuzu. Toward retail product recognition on grocery shelves. In *International Conference on Graphic and Image Processing*, 2015. doi: 10.1117/12.2179127