CLASSIFICATION OF GROCERY ITEMS USING CONVOLUTIONAL NEURAL NETWORK

CSE 676 DEEP LEARNING, FALL 2017
University at Buffalo
Project 1 Report

Aditya Pravin Walke
awalke@buffalo.edu (5020-5384)

Sunil Kunjappan Vasu
sunilkun@buffalo.edu (5020-5673)

Abstract

Recent advances in computer vision and its application has been applied to many fields. With the increase in performance of the machine learning technique in the past few years has helped in this advancement. One of the interesting domain is domestic environment, where a robot needs to detect and recognize a large variety of items which include grocery and food items. However this is a very challenging task due to the huge similarity between different items across the classes of grocery items. In this project we try to understand how the performance varies when different approaches to learn the grocery items. We use the Freiburg Groceries Dataset consisting of 5000 images spread across 25 classes are used for the training. Standard model like CaffeNet, Inception V4 and MobileNet are adopted for training and performance measure. We have created a ConvNet from scratch which is then trained on the dataset and the performance was measured. On an average we obtained a performance of ~65%. Data Augmentation technique was also used to understand how it affects performance for each of the model. This report illustrate the details of our approaches for classification of Freiburg groceries item dataset.

1. INTRODUCTION

In this project we aim to use Deep Convolutional Neural Network (CNN) for identification of Grocery items. This project is inspired by the *Amazon Go* which is a new kind of store featuring the world's most advanced shopping technology. The technology which allows customer to walk in and purchase items and walk out. This is a fusion of sensor system, Computer Vision and Deep Learning which makes the “Grab and Go” and “Just Walk out” possible. We tried multiple models like Inception V4, MobileNet[4], CaffeNet & normal CNN on the dataset provided by University of Freiburg.

2. TYPE OF TASK AND TASK DESCRIPTION

The main task of this project is classification of grocery items. We have a multiclass classifier which is trained to classify the grocery item into 25 different classes. Different models were selected to perform the training and to study the performance.

3. DATASET

The dataset used is The Freiburg Groceries Dataset [1] which consist of the grocery items images divided into 25 classes. Each of the classes has around 150 images. The total number of images are 4,973. These images where processed into training and validation sets and the test are run to find its performance.

Fig. 1: Example of Images from Freiburg Groceries Dataset

This is a challenging dataset as most of the grocery items look similar across classes and the model needs to be trained intensively to learn minute feature which helps to classify the items correctly. Some of the inevitable errors that we faced are the misleading images on the grocery items. For example a package of cereal may feature image of fruits and the model would associate this particular image to a Juice category as image of fruits are usually associated to the class Juice. Due to this inherent limitation, the model has to be fine tuned and trained on a large dataset which is created by data augmentation techniques.

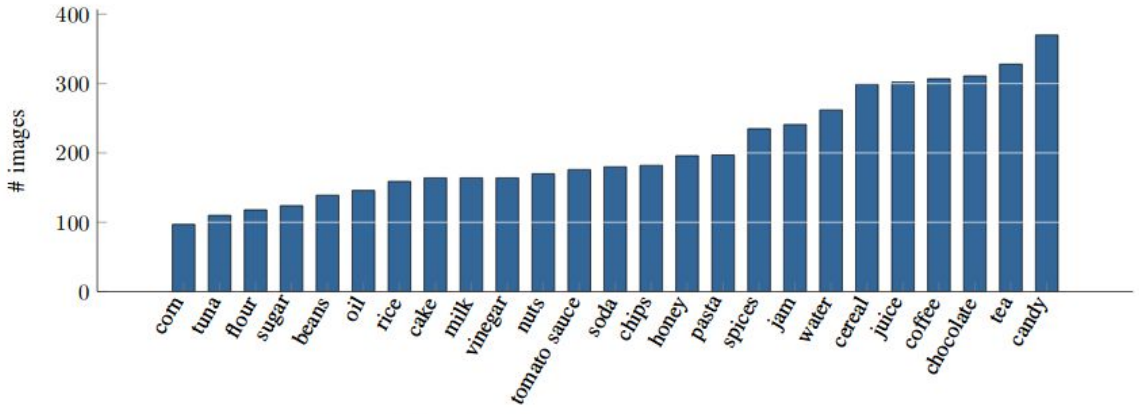


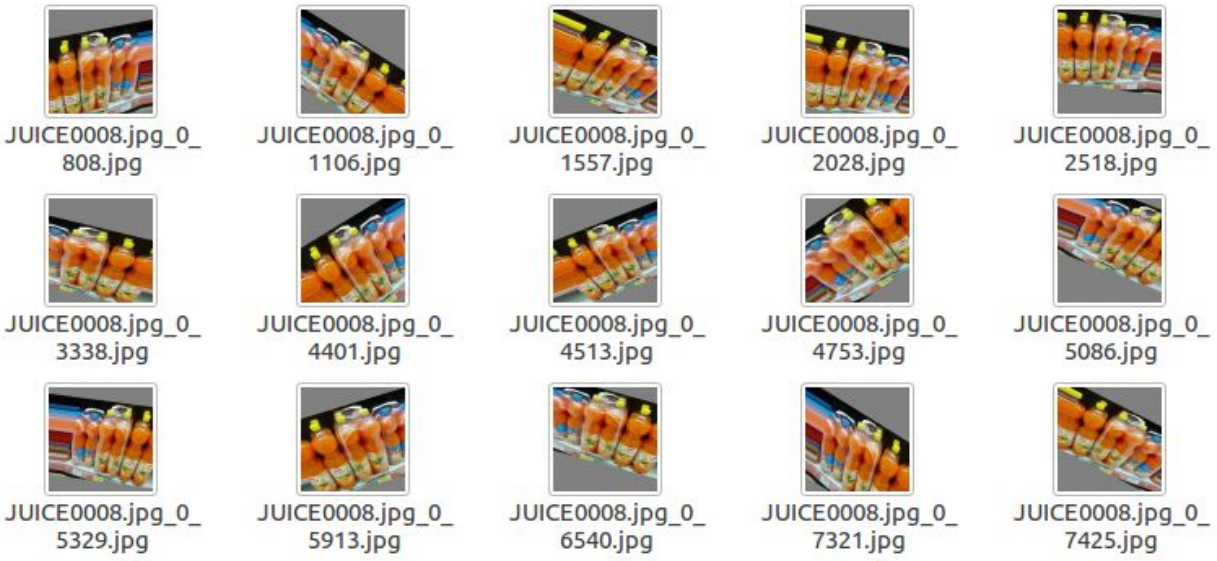
Fig. 2: Number of images per class in our dataset.

For the ease of computation we have created multiple subset of the main dataset which consist of 10 classes - Toy10, 5 classes - Toy5 and 3 classes - Toy3 respectively.

Fig. 3: Example of misclassification which highlight the challenges in our dataset, e.g., the model confuses a cereal boxes with drawings of fruits as juice.

We have created a new augmented dataset for the training purpose. This dataset contains additional ~20 augmented images for a particular image (82,481 in total), hence this is a very large dataset. Augmentation with regards to the flip, size, shear , zoom, rotation are done.

An original Image from Juice class



Corresponding Augmented images for an original juice image presented above

Fig. Example of Data Augmentation.

4. MACHINE LEARNING MODELS & METHODS USED

The approaches we used in this project are:

1. Recreate the Machine Learning CaffeNet architecture which was deployed by authors of the paper who published the Freiburg Groceries Dataset[5].
2. Transfer Learning and HyperParameter Tuning on Inception V4 to fit the dataset.
3. Transfer Learning and HyperParameter Tuning on MobileNet[4] to fit the dataset.
4. Created a Convolutional Neural Network Model from scratch which is then trained and tested on the Freiburg Groceries dataset.

The machine learning model used in the original paper The Freiburg Groceries Dataset[1] is a retrained model of CaffeNet. Our initial objective was to recreate the model from the original paper. We were able to successfully implement the Machine Learning model and were able to train and test the model using AWS.

The second objective was to adopt similar approach to create a model which can classify the grocery item. We have implemented additional two machine learning models which performs Transfer Learning on Inception V4 and MobileNet and experimented some hyperparameter tuning.

Inception V4 model is tuned and fitted to our dataset. The Inception architecture that has been shown to achieve very good performance at relatively low computational cost which is the deciding factor for selecting this model. Inception V4 is trained on 1000 everyday object categories (ImageNet ILSVRC). Kera and TensorFLow was chosen as the platform for creating our model. **Keras** is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. Keras helps us to focus on enabling fast experimentation.

However the already trained Inception V4 does not perform well on our model as it has not trained on the classes we are interested in. Initial testing on the Kera Inception model shows that majority of the items gets classified as “Grocery” or “Food” which is not least informative to us. Hence this model needs to be trained on the dataset specific to our needs.

It’s well known that CNN training require significant amounts of data and resources to train. For example, the ImageNet ILSVRC model was trained on 1.2 million images over the period of 2–3 weeks across multiple GPUs. Hence creating a model from scratch is a very cumbersome and time consuming task. So in this situation we adopted transfer learning and hyperparameter tuning. These methods can be summarized as follows:

1. **Transfer Learning**: The last fully connected layer of a pre trained CNN is removed and treated as a feature extractor for the new dataset. Once we have successfully extracted all the features for all images we train a classifier on the new dataset.
2. **Hyperparameter-tuning:** These are the parameters that are not learnt by the network on its own, adjusting these parameters according to the specific needs of the network is essential in improving the performance.

Using these two methods the model is once again trained on The Freiburg Groceries Dataset[1]. The training are done of AWS g2.2xlarge[3] GPU for faster computation.

To better improve the model and to overcome overfitting due to the lack of images we tried to leverage Data Augmentation. We created multiple variation images of the same image with different image property. We hoped this would enhance the learnability of the model.

The next approach was to use the technique of transfer learning and hyperparameter tuning on MobileNet. This approach is to understand how the performance varies when a different model is used for transfer learning. The principal difference is that Inception V4 is optimized for accuracy, while the MobileNet are optimized to be small and efficient, at the cost of some accuracy. There was significant improvement in training when using a MobileNet.

The final objective was to create a model of our own and understand its performance. We tried creating a simple convolutional neural network by varying the number of convolutional layers, pooling layers and fully connected layers. The created model was run on AWS and the performance was evaluated with the other models.

5. PERFORMANCE MEASURE AND RESULTS

Due to the limitation of training the model because of the limited computational power available for us we were not able to explore the complete performance of the model. Once the model is trained on all the dataset (including the augmented data) the accuracy can be improved significantly.

The Original Model from the paper was trained and it obtained an accuracy of 81.6%.

```
I1016 23:10:26.504088   939 data_layer.cpp:73] Restarting data prefetching from start.
I1016 23:10:37.544456   935 solver.cpp:447] Snapshotting to binary proto file ../results/0/snapshots/_iter_10000.caffemodel
I1016 23:10:38.739011   935 sgd_solver.cpp:273] Snapshotting solver state to binary proto file ../results/0/snapshots/_iter_10000.solverstate
I1016 23:10:39.434528   935 solver.cpp:310] Iteration 10000, loss = 0.00674226
I1016 23:10:39.434581   935 solver.cpp:330] Iteration 10000, Testing net (#0)
I1016 23:10:40.100881   940 data_layer.cpp:73] Restarting data prefetching from start.
I1016 23:10:40.682293   935 solver.cpp:397]     Test net output #0: accuracy = 0.816191
```

Fig. CaffeNet Result

The Re-Trained Inception V4 was able to give a performance of 63.5%. Inception V4 was trained on a Toy3 dataset which consist of 3 classes. The training of Inception on complete dataset on AWS was cumbersome and we were not able to complete as the computation stalls indefinitely. The re-trained MobileNet was able to outperform the Inception V4 in case of performance as well as the time taken for computation. MobileNet was trained on the complete dataset.

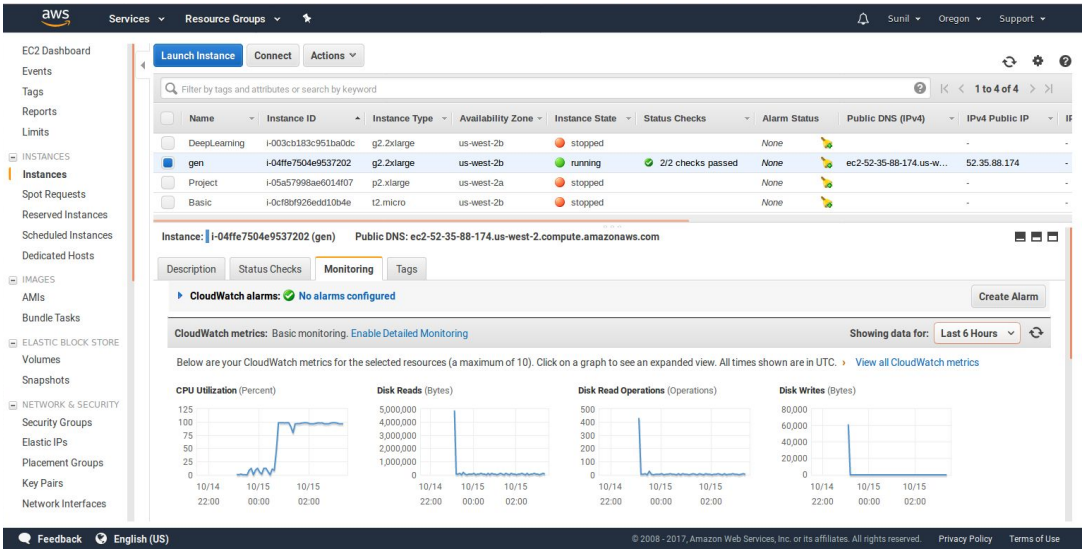


Fig. Inception V4 Training stalling on AWS.

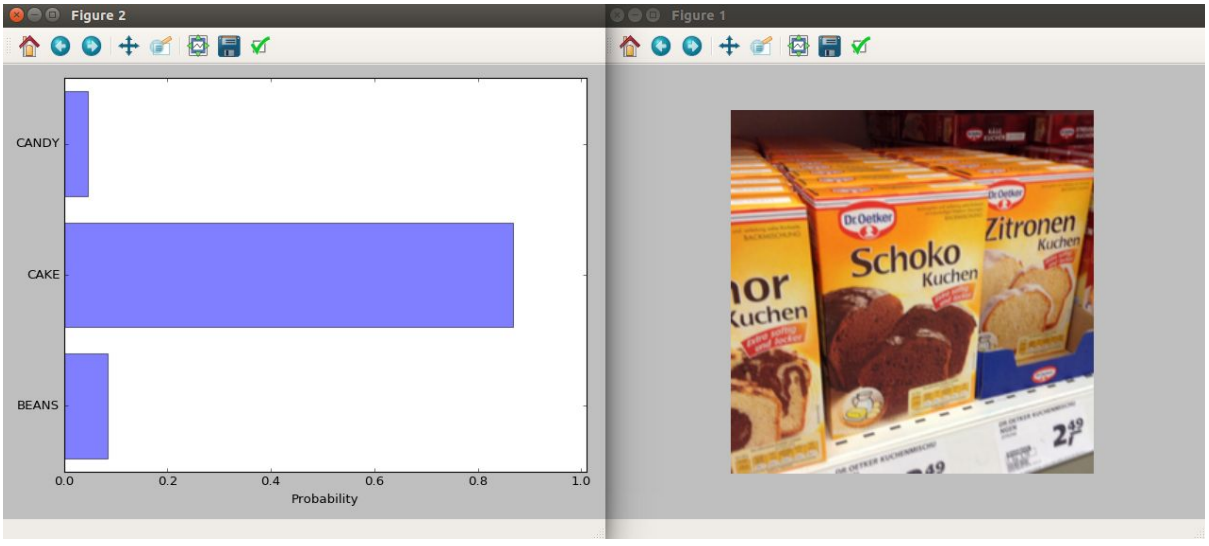


Fig. Suggested output of Inception V4 model for a image from class cake

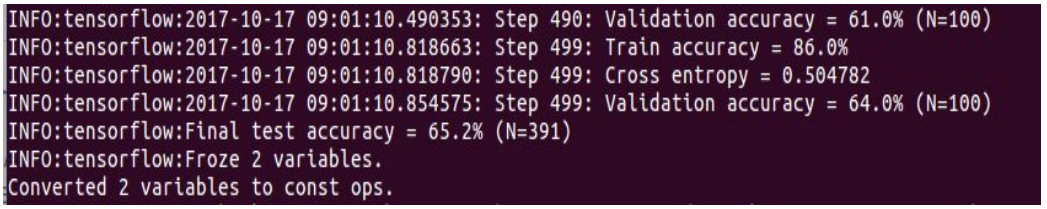


Fig. MobileNet Training on Complete Freiburg Groceries Dataset.

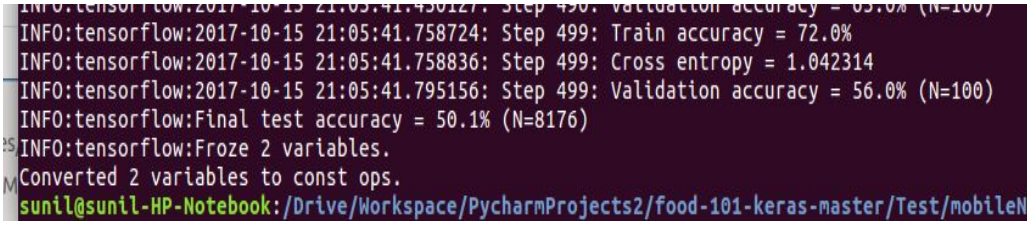


Fig. MobileNet training on Augmented Dataset

Fig. Suggested Results for an image of Water by MobileNet

The customized ConvNet we created was able to give a performance of 22.6%

Fig. Simple Convolutional Neural Network Result

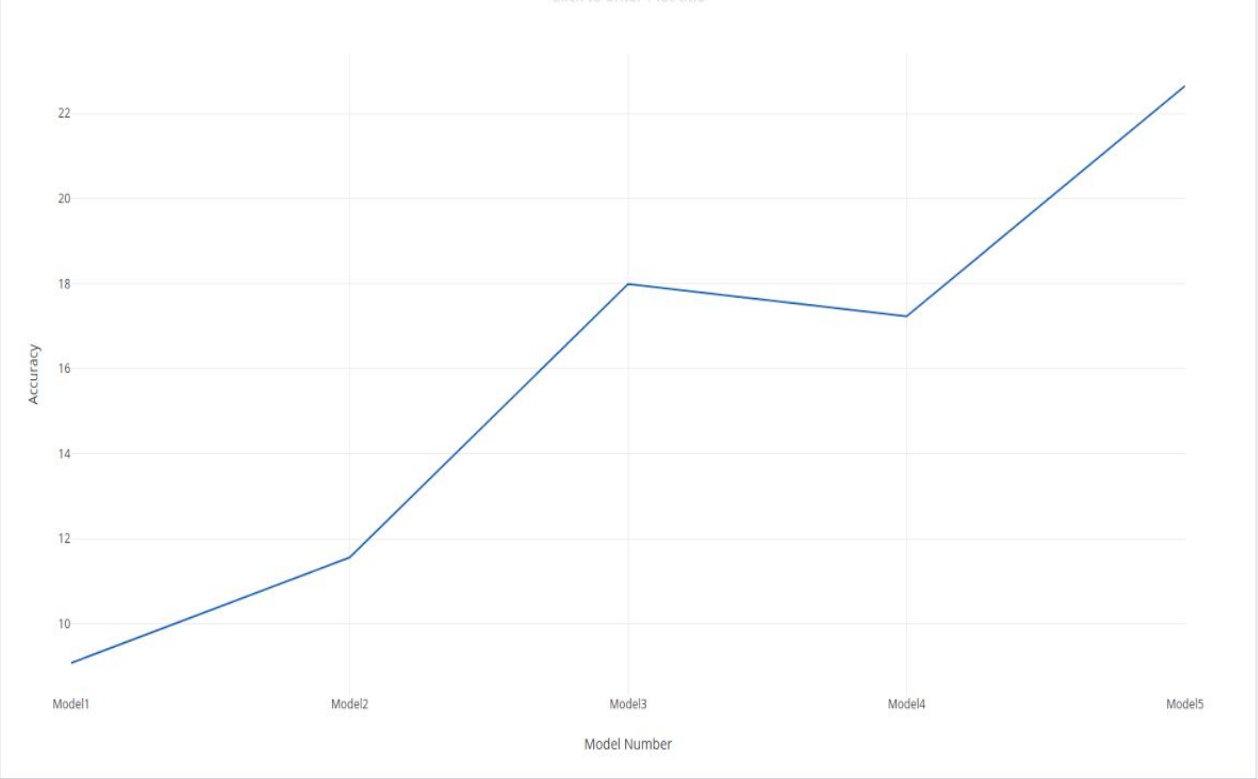


Fig. Model Performance

Model Number	Model Details
1	2 Convolutional Layers followed by Max Pooling Layers, 1 Dropout layer & Fully Connected Layer.(500 iterations)
2	2 Convolutional Layers followed by Max Pooling Layers, 1 Dropout layer & Fully Connected Layer.(20000 iterations)
3	3 Convolutional Layers followed by Max Pooling Layers, 2 Fully connected Layers(20,000 iterations)
4	3 Convolutional Layers followed by Max Pooling Layers, 2 Fully connected Layers and 1 Dropout Layer(10,000 iterations)
5	5 Convolutional Layers followed by Max Pooling Layers and normalization Layers, 3 Fully connected Layers and 2 Dropout Layers (25,000 iterations)

The final result from all the training can be summarized as follows.

Performance Measure	
Original Model CaffeNet - AlexNet adaptation	81.6% (Complete Dataset)
Re-Trained Inception V4	63.5% (Toy3)
Re-Trained MobileNet	66.2% (Complete Dataset)
Customized ConvNet Model	22.6% (Complete Dataset)

#Toy3: Scaled down Freiburg Groceries Dataset with 3 classes.
#Complete Dataset: The complete Freiburg Groceries Dataset with 25 classes.

6. CONCLUSION AND SCOPE FOR IMPROVEMENT

In this project we have used different approaches to model a grocery item classification CNN. We have used pre-trained Inception V4 and MobileNet, on which transfer learning and fine tuning was performed to work with the Freiburg Groceries Dataset. As well as transfer learning through CaffeNet and implementation of a simple CNN was tested. Limitation was faced in this project in terms of computational resources which forced us to train on smaller dataset or use a simpler model which lead us to a situation where we were not able to train and understand the complete potential of our models. The main reason for not achieving a high recognition accuracy is due to the nature of the dataset where all items are very similar to each other and very fine feature extraction must take place to gain a proper classification of the items. For such fine feature extraction much more complex models are required which demand high computational resource and time.

As a future direction, we aim at recognizing food items in images with higher accuracy. We aim to explore the options of getting a better accuracy even with a smaller network with a better understanding of the dataset. Further investigation on transfer learning and fine tuning are to be done. Better understanding of data augmentation and how it affect the accuracy needs to be studied.

7. REFERENCE

[1] https://arxiv.org/pdf/1611.05799.pdf
[2] https://arxiv.org/abs/1602.07261
[3] https://aws.amazon.com/ec2/instance-types/
[4] https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/#0
[5] http://ais.informatik.uni-freiburg.de/publications/papers/jund16groceries.pdf
[6]https://deeplearningsandbox.com/how-to-use-transfer-learning-and-fine-tuning-in-keras-and-tensorflow-to-build-an-image-recognition-94b0b02444f2