Image classification for e-commerce

Nestor Pereira

May 2020

1 Introduction

The world of e-commerce is revolutionizing how we shop and allowing us to have access to new merchandise and diverse products. But it also brings some problems, such as too many products and information out there, making it hard to choose the desired one that will meet our needs. Many online retailers have specialized on certain product types or areas, decreasing the number of products and facilitating the decision for the customer. While, others due to their high demand of products in every area, had to find other solutions. Recommender systems is a common solution where products are recognized, grouped and then offered as alternatives to similar products searched by the customer.
In the clothing industry, recommender systems are very much selected by type of clothing and many algorithms that classify clothes into different groups. Clothes can be categorized in different groups based on their shape, use, color and even season.

2 Background

Several machine learning classification methods have been applied clothing images to label them in type of garment, color, gender, use and season.

Convolutional neural network have show a high accuracy for images classification problems. Several CNN for classification have been used in the famous "ImageNet Large Scale Visual Recognition Challenge (ILSVRC)" competitions, which evaluates algorithms for object detection and image classification at large scale.

In the next graph we show how the error for the Image net data set has improved over the years with different CNN classifiers.

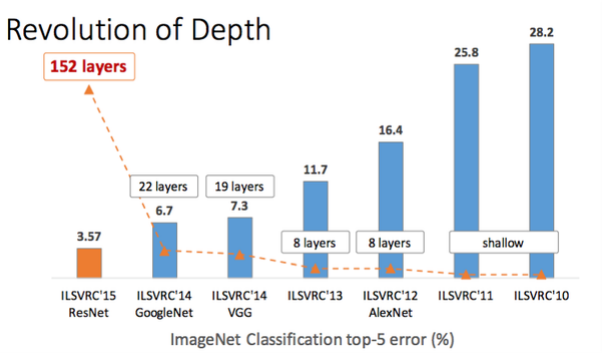


Figure 1: Backward filtered state prediction vs original states

There are two types of clothing images used in classification papers. No background and background ones, where the background ones usually include a human wearing the clothes and can get very tricky since more than one garment can appear in the picture making the classification more complicated.

Data augmentation can help solve this problem. Random rescaling and cropping can help understand what item we are trying to label. Basically by random cropping part of the images and doing label classification, we will get an item label better then the rest. This usually means that the object of interested is usually the one the image is focused on and has the easiest classification.

3 Data

After reading [1], found that they use the fashion MNIST dataset as an example, therefore we are going to take advantage of that to help us understand the basics of a NN classifier and use their code as a benchmark for out new dataset. We are going to study the fashion MNIST

dataset where 70000 28x28 labeled clothing images are provided. As a starting point, we are going to train our classifiers with 60.000 images and test out models with 10.000 images
The data set has different 10 classes:

$Tshirt/top|Trouser|Pullover|Dress|Coat$
$Sandal|Shirt|Sneaker|Bag|Ankleboot$

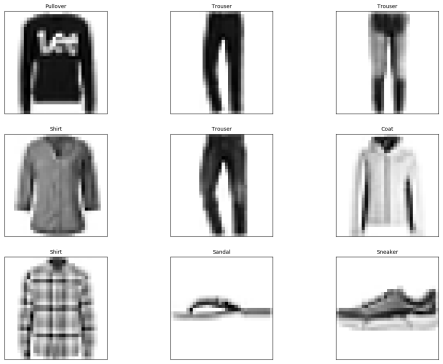


Figure 2: MNIST no background data set

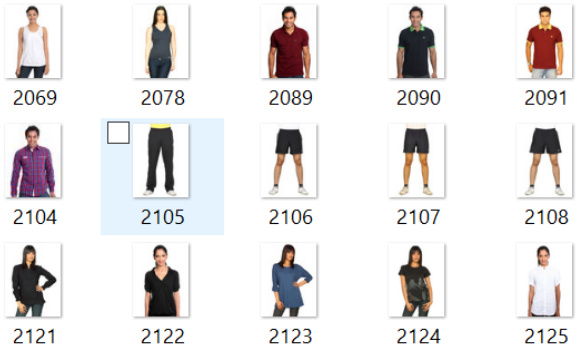


Figure 3: Kaggle Background + human data set

Index	id	gender	articleType	baseColour	season	usage
0	1163	Men	Tshirts	Blue	Summer	Sports
1	1164	Men	Tshirts	Blue	Winter	Sports
2	1165	Men	Tshirts	Blue	Summer	Sports
3	1525	Unisex	Backpacks	Navy Blue	Fall	Casual
4	1526	Unisex	Backpacks	Black	Fall	Sports
5	1528	Men	Jackets	Black	Fall	Sports
6	1529	Men	Tshirts	Red	Fall	Casual

Figure 4: Kaggle image labeling

Next step is to find a more complicated dataset with high resolution labeled clothing images with a background to make things more interesting.

A more complicated dataset was found, cleaned and image resized. Clothing Kaggle dataset [2] contains 44447 images with different resolution between 60X80 and 80x80 pixels. They are all going to be resized to 60x80 pixels. The problem with this dataset is that classes are unbalanced, making the fitting of the model hard. Tried running it with simple NN classifier and the outputs were just constant scores for all the testing samples, which is caused by an unbalanced dataset. The cleaned and resized dataset will be included in the files for future use.

4 CNN classifiers

Convolution neural networks have shown a better performance specially in computer vision. Several CNN architectures are going to be tested on both datasets and compared with different performance measures

4.1 NN sequential model (Benchmark)

Found in [1] a benchmark model for classification of 28x28 images which consist in 2 fully connected dense layers and softmax output activation layer.

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_3 (Flatten)          (None, 1024)              0
_________________________________________________________________
dense_9 (Dense)              (None, 300)               307500
_________________________________________________________________
dense_10 (Dense)             (None, 100)               30100
_________________________________________________________________
dense_11 (Dense)             (None, 10)                1010
=================================================================
Total params: 338,610
Trainable params: 338,610
Non-trainable params: 0
```

Figure 5: NN benchmark model structure

4.2 Lenet-5

Lenet5 was created for the famous MNIST digit data set label classification. Images were resized to 32x32. The model architecture is show below:

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 30, 30, 6)         60
_________________________________________________________________
average_pooling2d_1 (Average (None, 15, 15, 6)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 13, 13, 16)        880
_________________________________________________________________
average_pooling2d_2 (Average (None, 6, 6, 16)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 576)               0
_________________________________________________________________
dense_2 (Dense)              (None, 120)               69240
_________________________________________________________________
dense_3 (Dense)              (None, 84)                10164
_________________________________________________________________
dense_4 (Dense)              (None, 10)                850
=================================================================
Total params: 81,194
Trainable params: 81,194
Non-trainable params: 0
```

Figure 6: Lenet 5 model structure

4.3 ResNet

This model was a novel architecture, differentiating in the fact that it could go deeper since it could skip connectors. One of its main features is batch normalization allowing the network to train more layers than the previous ones. The architecture using keras is show below:

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dropout_1 (Dropout)          (None, 2048)              0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                20490
=================================================================
Total params: 23,608,202
Trainable params: 23,555,082
Non-trainable params: 53,120
```

Figure 7: modelresnet

5 Results

Our classifier prediction outputs, for each test observation, an array of probabilities of an object belonging to a class. Higher probabilities will show to what class an object belongs to.

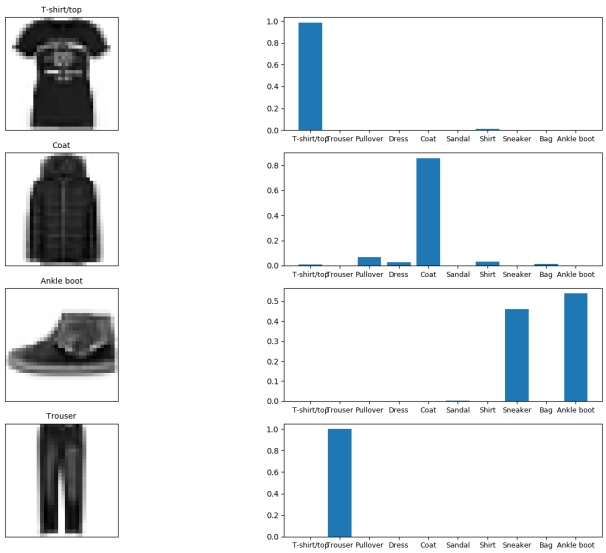


Figure 8: modelresnet

5.1 Performance numeric measures

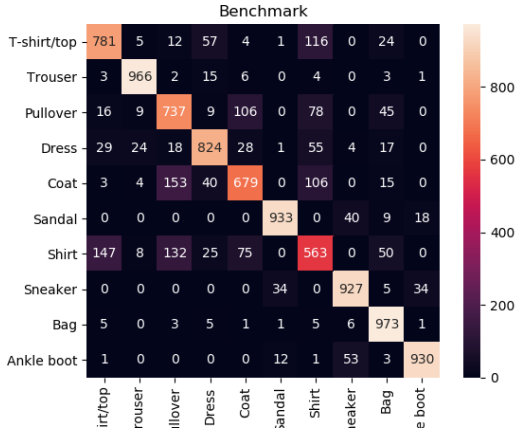
Some classification measures have been used to quantify our predictor's behaviour. Classification accuracy is improved when more complex classifiers are used. Going

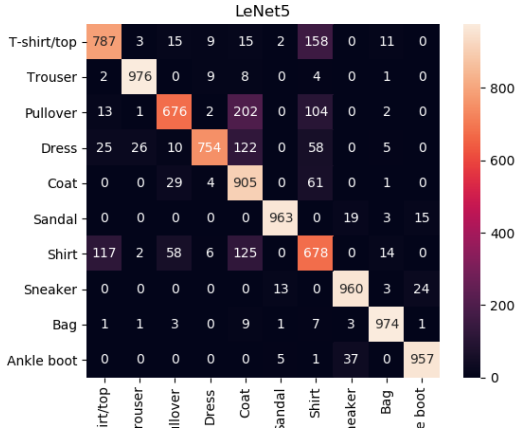
from the basic NN benchmark to the Lenet then to the Resnet.

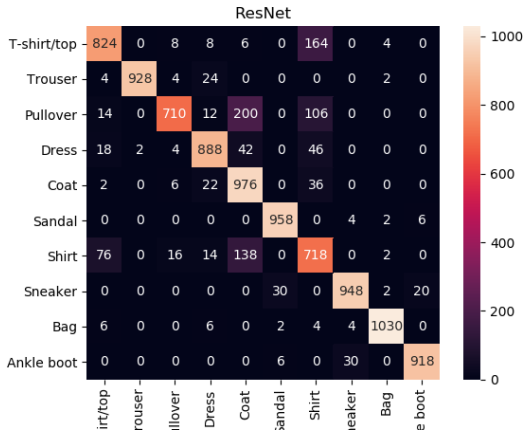
I'm concerned with my ROC auc results, since they are too high for the accuracy levels obtained for all the classifiers. Looking at the curves shapes and values, they look correct, but the calculation of the Area under the curve returns a non correct value. I'm going compare the classifiers visually based on the shape of the curves more than the numeric results

Table 1: Numeric results

Classifier	Accuracy	Precision	ROC_auc
Benchamrk	0.816	0.92	0.98
LeNet 5	0.86	0.94	0.99
ResNet	0.89	0.96	0.99







5.2 Confusion matrix

Confusion matrixes show the number of correctly classified observations and incorrect ones. We can see that there is trouble for differentiating top clothes such as Shirts, T-shirt, pullover and coats. We can see more correct classifications on the diagonal of the matrix as we increase on the complexity of the classifier.

5.3 Precision and recall

Studying the precision and recall curves, we can see that more complex classifiers, have curves closer to the top right of the plot with higher precision and recall.

$$\text{Precision} = \frac{\text{Num of Relevant Images Retrieved}}{\text{Num of Retrieved Images}} = \frac{\text{TP}}{\text{TP+FP}}$$

Figure 9: precision

$$\text{Recall} = \frac{\text{Num of Relevant Images Retrieved}}{\text{Num of Relevant Images}} = \frac{\text{TP}}{\text{TP+FN}} = \text{Sensitivity}$$

Figure 10: recall

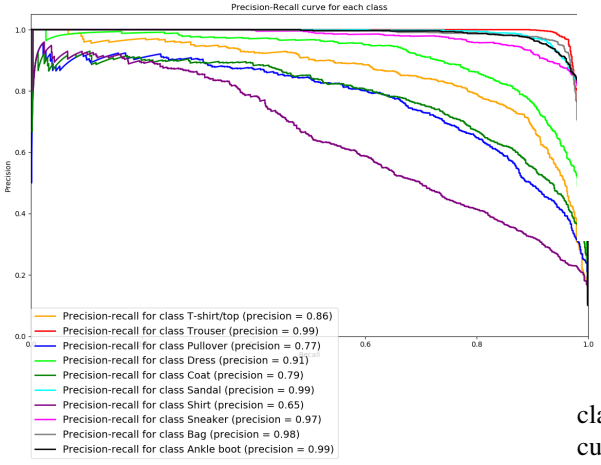


Figure 11: NN benchmark model

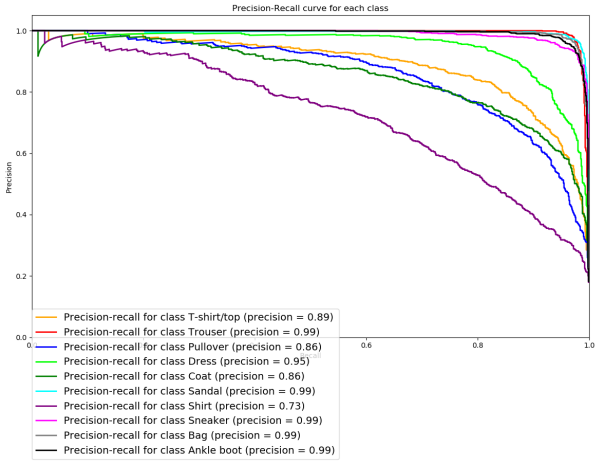


Figure 12: Lenet 5 model

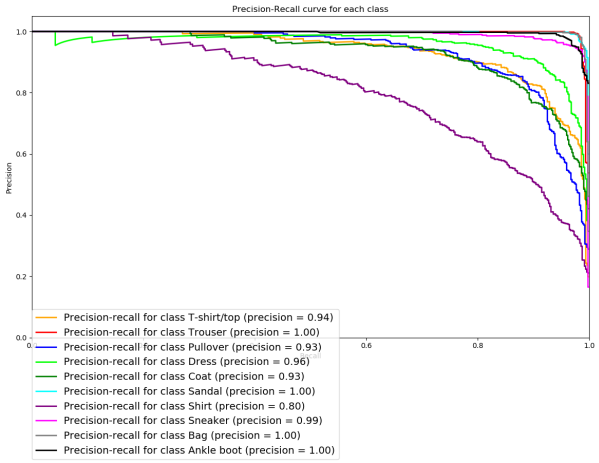


Figure 13: model resnet

We also binarized the output for correct and incorrect classifications and ploted the ”total” precision and recall curve for each classifier, where we can see that the deeper and complex the classifier is, the better precision and recall we get.

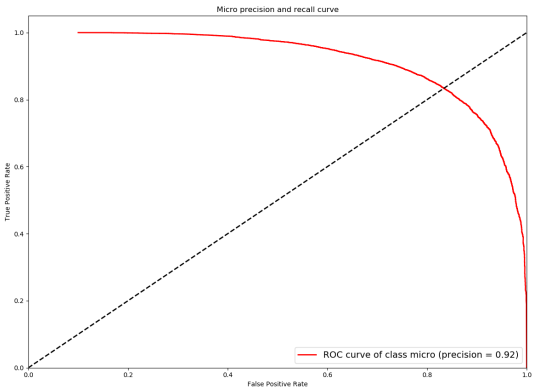


Figure 14: NN benchmark model

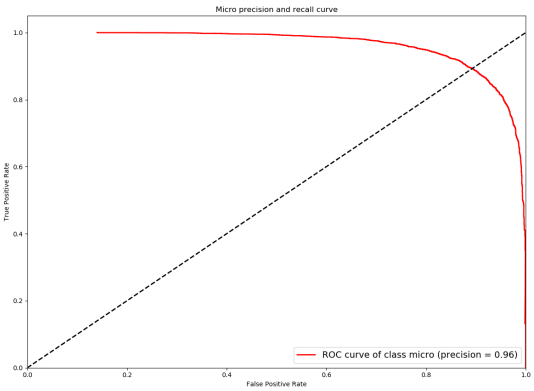


Figure 16: model resnet

5.4 ROC curves

Looking at the ROC curves, we can see that more complex classifiers, have curves closer to the top left of the plot with higher True positive rate and lower false positive rate.

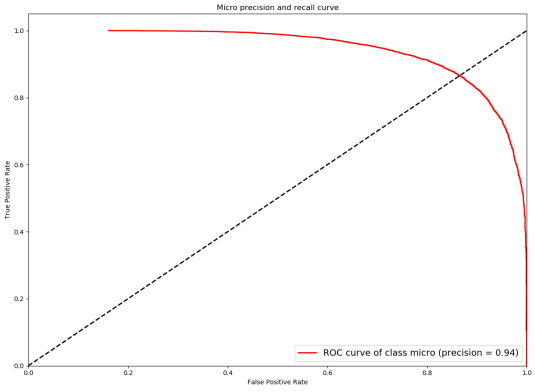


Figure 15: Lenet 5 model

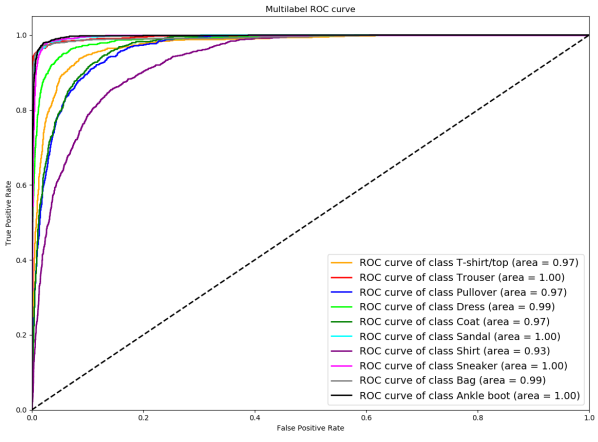


Figure 17: NN benchmark model

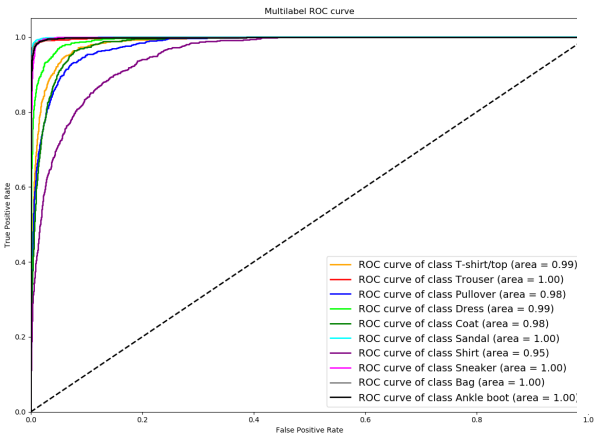


Figure 18: Lenet 5 model

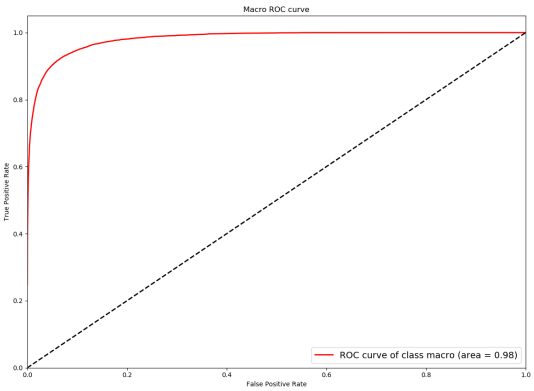


Figure 20: NN benchmark model

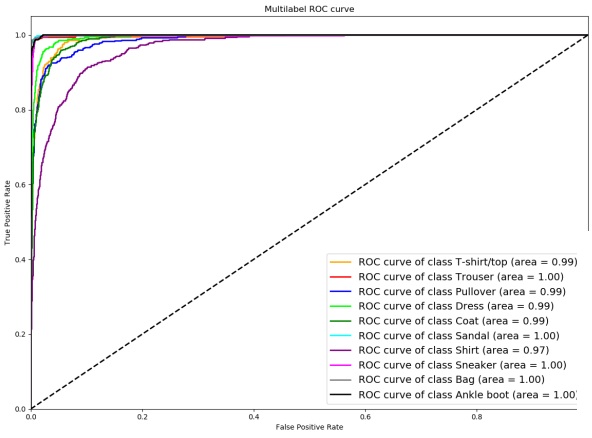


Figure 19: model resnet

After binarizing the output for correct and incorrect classifications and ploted the ”total” ROC curve for each classifier, where we can see that the deeper and complex the classifier is, the higher true positive rate we get and the lower False positive rate.

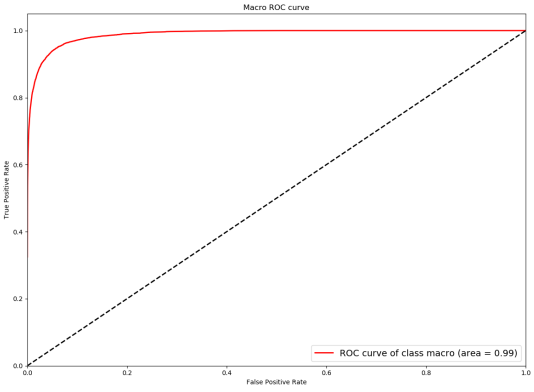


Figure 21: Lenet 5 model

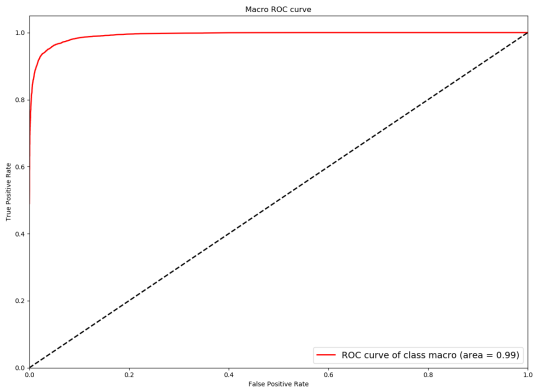


Figure 22: model resnet

6 Conclusion

The objective of this project is to accurately label grayscale small images into 10 classes, compare the accuracy of different classifiers for each label and all together. A deep learning approach have been selected where a basic NN, LeNet5 and Reset have been implemented and compared.

The analysis showed that ResNet was the best model. Also some labels such as Trousers, dress, sandals, sneakers and bags were easier to label, while other like Tshirts, shirts, pullovers and coat would get mixed in some occasions.

6.1 Future work

This study was using no background grayscale images, which are easier to accurately classify.

RGB image classification would be a future step to follow. It wasn't implemented in this project because lack of hardware memory for running deep neural networks for 3 layer colored images.

Non background images could be obtained from the already clean Kaggle data set and run on this same classifiers.

Data augmentation could be use to increase the size of out training set to increase the accuracy.

7 References

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow : Concepts, Tools, and Techniques to Build Intelligent Systems

2. https://www.kaggle.com/paramaggarwal/fashion-product-images-datas

3. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

4. A. Schindler, T. Lidy, S. Karner, and M. Hecker, "Fashion and apparel classification using convolutional neural networks," arXiv preprint arXiv:1811.04374, 2018.