Machine Learning and Content Analytics

Author: Bellos Adam, f2822009

March 2022

Contents

Introduction

Fashion industry is a perpetual and constantly evolving sector of the economy. Whether sold online or in a physical store, clothing sales constitute a significant proportion of the global GDP. The significance of the industry in today's economy is depicted by market insights. Research shows that the industry is valued at over 3 trillion dollars and responsible for 2% of the global GDP.

Technological and digital innovations during the fourth industrial revolution have revolutionize the way companies operate. Fashion industry is not an exception, automation and data-driven solutions generated by artificial intelligence (AI) or machine learning are changing every aspect of this forward-looking business domain. Companies in the industry from producers, suppliers to retailers are integrating these new technologies to remain relevant in a highly competitive marketplace. Notably, nowadays 44% of the fashion retailers who have not adopted AI technologies are facing bankruptcy. As a result of this, global spending on AI technologies by the fashion and retail industry is expected to reach $7.3 billion by the end of the year.

To better understand how important such technologies are, such as mechanical learning and artificial intelligence for the fashion industry we will refer to some uses they see today. In order to take advantage of the high volume and the availability of data generated daily, fashion brands implement these technologies to understand customer needs and design better apparel. Nowadays, designs are based on customer's preferred colors, textures, features and other style preferences. In addition, fashion companies have become quicker in providing instant gratification to their consumers by understanding seasonal demands and manufacturing the right supply of specific products. All this is made possible by the use of technologies such as chatbots, machine learning algorithms, computer vision and deep learning. In short, industry 4.0 is transforming how fashion enterprises are designing and manufacturing their products as well as how they are marketed and shipped to the customer.

Project, Vision and goals

This work focuses on one aspect of the use of artificial intelligence in the fashion industry, which is multiclass classification using a deep learning approach. Image classification is the task of categorizing and assigning labels to groups of pixels or vectors that represent images. The main functions of an Ai-powered classifier include understanding consumer behavior, identifying advertising trends on social media, customer engagement, promotions and much more. All these activities are of vital importance for various segments within the fashion industry such us e-commerce, retail, traditional and digital marketing.

Many other traditional classification models have been trained on this data set. A secondary goal of this project, beyond the development of the classifier, is to see if a deep learning model would be more efficient and precise compared to conventional classification logarithms.

Data Collection and Dataset Overview

The dataset used for the purpose of this project was created by Han Xiao, Kashif Rasul and Roland Vollgraf [1]. The methodology behind the creation of the dataset along with its description was published on 2017 in a paper with the title "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms". The dataset is based on Zalando's fashion item pictures. Zalando is a fashion platform based in Germany that takes advantage of artificial intelligence and data-driven insights to serve its customers.

To build the Fashion-MNIST dataset 70.000 unique front look fashion images were used. The actions that took place for the creation of the dataset, as described in the aforementioned paper, are as follows:

1. The (51 × 73) JPEG pictures were converted in PNG.
2. Edges that were close to the color of the corner pixels were trimmed.
3. The longest edge of the image was resized to 28 by subsampling.
4. The pixels were sharpened using a Gaussian operator of the radius and standard deviation of 1.0, with increasing effect near outlines
5. The shorter edge of the image was converted to 28.
6. Negation of the color intensity
7. Image was converted to 8-bit grayscale pixels.

The resulted dataset, which has been used for this project, is comprised by 70,000 fashion products in 28 x 28 grayscale format. These products come from all gender groups and belong to 10 different types of clothing. The product classes are T-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag and ankle boot. Each image has been labeled with a unique number 0-9 which differentiate each class of product. The mapping of all 0-9 integers to class labels can be seen in the table below. Furthermore, the dataset contains 7,000 images of each class.

Label	Class
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

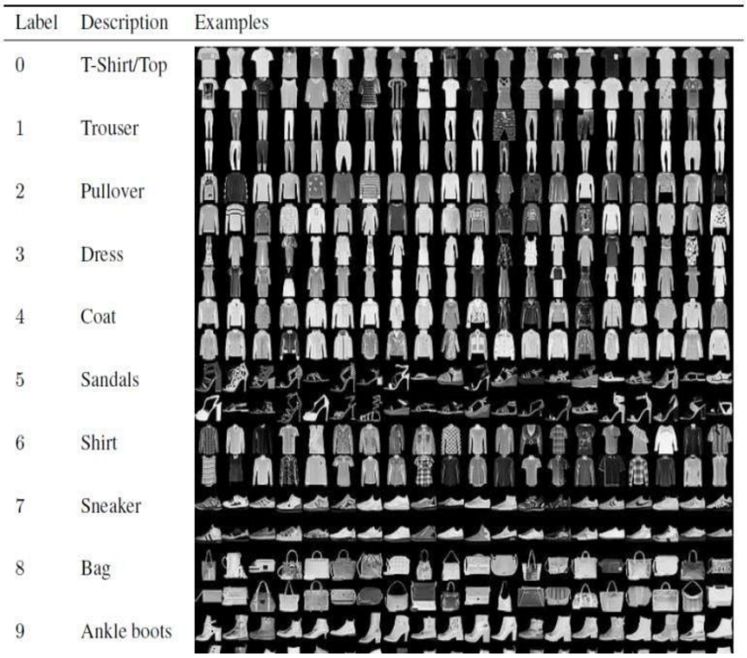

Figure 1. Fashion-MNIST dataset

Algorithms, architectures/systems

Image classification techniques are mainly divided into two categories: Supervised and unsupervised image classification techniques.

An **unsupervised** classification algorithm is a fully automated process that does not leverage training data. This means machine learning algorithms are used to analyze and cluster unlabeled datasets by discovering hidden patterns or data groups within the input dataset. A well-known and frequently used unsupervised method is the K-means algorithm that groups objects into k groups based on their characteristics.

Meanwhile, a **supervised** image classification algorithm uses previously classified reference samples, commonly referred as the ground truth, in order to train the classifier and subsequently classify new, unknown data (images that the model has not been trained on). A representative of this group is the decision tree classifier. Its architectures resemble that of a tree with root, branches and leaves. To predict a class label for a record we start from the root of the tree. The root consists of the entire population which is then divided into two or more homogeneous sets. Then we compare the values of the root attribute with the record's attribute. Based on that comparison we follow the branch corresponding to that value and jump to the next node. The process goes on until we reach a leaf node which provides the classification of the record.

In recent years, the increase of computing power and the continuous research in the field of artificial intelligence brought to the fore deep learning classification techniques. Among them, the **convolutional neural network (CNN)** has demonstrated excellent results in computer vision tasks, especially in image classification. Convolutional Neural Network is a special type of multi-layer neural network inspired by the mechanism of the optical and neural systems of humans. A CNN is able to learn and train from data on its own without the need for human intervention. CNN's popularity in image classification tasks is derived by it's high performance and its easiness in training. A typical CNN is comprised by two parts, the convolutional base and a classifier (fig 2). The convolutional base is composed by several convolutional and/or pooling layers. Its aim is to generate features from the images, features that are used by the classifier to classify the images of the given dataset.

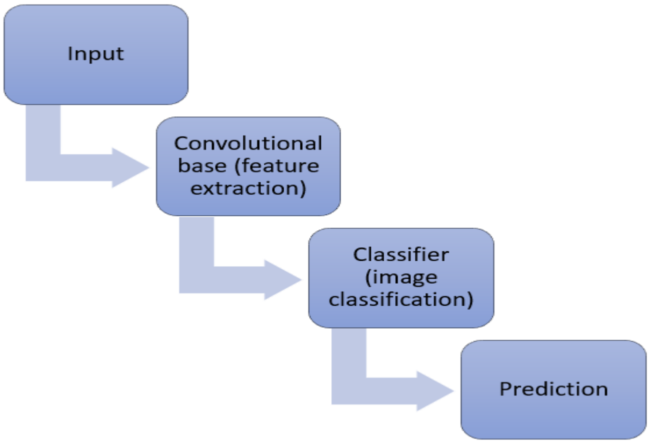

Figure 2. Parts of CNN

Another advantage of a CNN is that it automatically learns hierarchical feature representations. This means that features computed by the first layers are more general and can be reused in different but similar tasks, while features computed by the last layers are specific and depend on the chosen dataset and task. This characteristic of the CNN enables the use of networks that have already been trained to solve a classification task of a different dataset. Such pretrained models are used by removing the original classifier and replacing it with a new classifier. Then, someone has to select which layers of the pre-trained model will be trained on the new dataset. There are three options:

- Train the entire model. Meaning all the layers.
- Train some layers and leave the others frozen
- Freeze the convolutional base.

For the purpose of our classification task three models were implemented and compared. One CNN (fig 3) was built from scratch and two pre-trained models [2] were used for a transfer learning solution. The first pretrained model used is ResNet50 which is trained on the ImageNet dataset. For this model I choose to train some of the last layers while the previous layers are left frozen. The second pretrained model implemented for our image classification task is ResNet18 which is also trained on ImageNet dataset. Although, in this case I followed a different approach as I choose to train the entire model. Both pre-trained models were imported from torchvision.

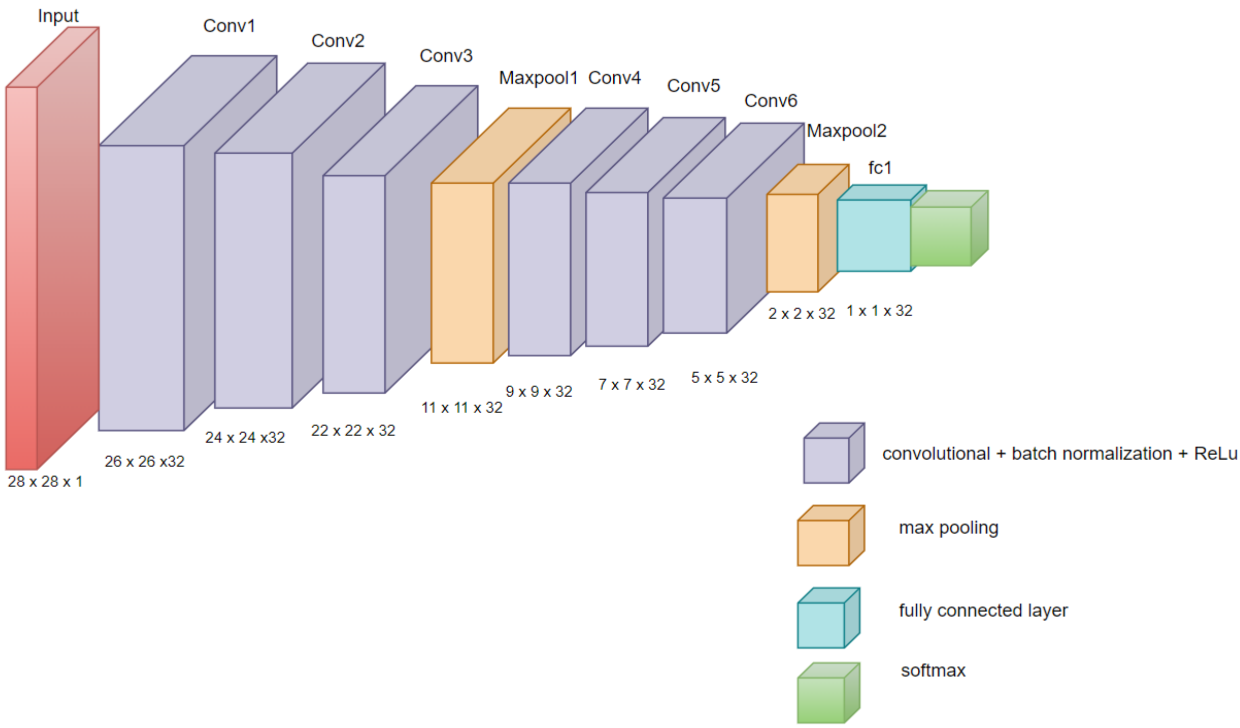

Figure 3. Architecture of the CNN.

Experiments – Setup, Configuration

For the purpose of the first experiment, 60,000 observations were used for training, 10,000 for validation. Both training and validation sets are balanced in terms of class representation. Each class of the training set consists of 6,000 records while each class of the validation set consists of 1,000 records (images). Furthermore, various parameters were defined in accordance with our resources and project goals. Some of these parameters can be seen below.

For the CNN model:

- Number of training epochs: 50
- Batch size: 128
- Learning rate: starting with 0.001. The learning rate is reduced by half every 20 epochs.
- Step size: 20
- Gamma: 0.5
- Optimizer: Adam

For ResNet50:

- Number of training epochs: 10
- Batch size: 128
- Learning rate: starting with 0.003
- Gamma: 0.1
- Optimizer: Adam

For ResNet18:

- Number of training epochs: 15
- Batch size: 128
- Learning rate: starting with 0.001. The learning rate is multiplied by 0.1 at the 4^{th},6^{th} and 8^{th} step
- Gamma: 0.1
- Optimizer: Adam

Results & Quantitative Analysis

In order to evaluate and compare our classification models, we made use of various popular performance metrics. We are going to explain them in more detail in the next paragraphs. Nonetheless, regardless of their differences, they all aim to answer a simple question "How well the model classifies an image".

The models we are going to compare are the developed CNN, ResNet50, ResNet18, decision tree classifier and a KNN classifier. The last two classifier were taken from the "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms" paper [1] and are chosen based on their performance on Fashion-MNIST data set. The models are compared based on their accuracy (fig4). Informally, accuracy is the fraction of predictions the model got right, meaning the number of images classified correctly. Formally, accuracy has the following definition:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

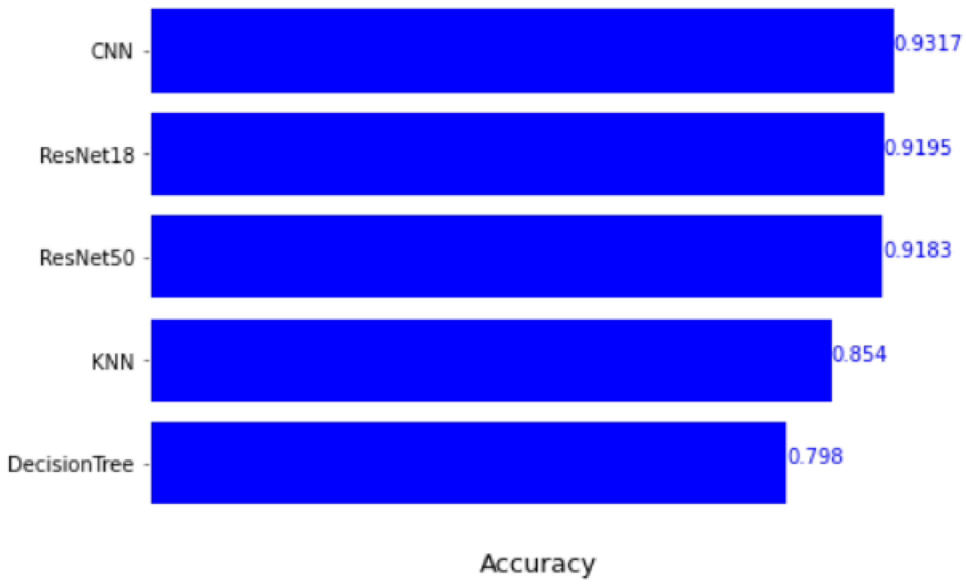

Figure 4. Model comparison in regard with accuracy

As it can be inferred models that are based on CNN architecture perform better as they classify more images correctly. Thus, we will focus on those and compare them further. Other metrics used for the comparison is precision, recall and F1 score (fig 5). Precision is calculated as the sum of true positives across all classes divided by the sum of true positives and false positives across all classes. Recall is defined as the number of true positives divided by the sum of true positives and false negatives. Meanwhile F1 score is derived from precision and recall.

$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN} \qquad F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})}$$

Model	Precision	F1_score	Recall
CNN	0.9266	0.926	0.9261
ResNet18	0.9190	0.919	0.9183
ResNet50	0.9180	0.918	0.9184

Figure 5. Precision, recall and F1 score comparison

From the model comparison we can see that ResNet18 and ResNet50 have similar performance while the CNN seems to perform better. To get a deeper understanding of how well CNN performs we can see the fraction of correct predictions per class on fig 6.

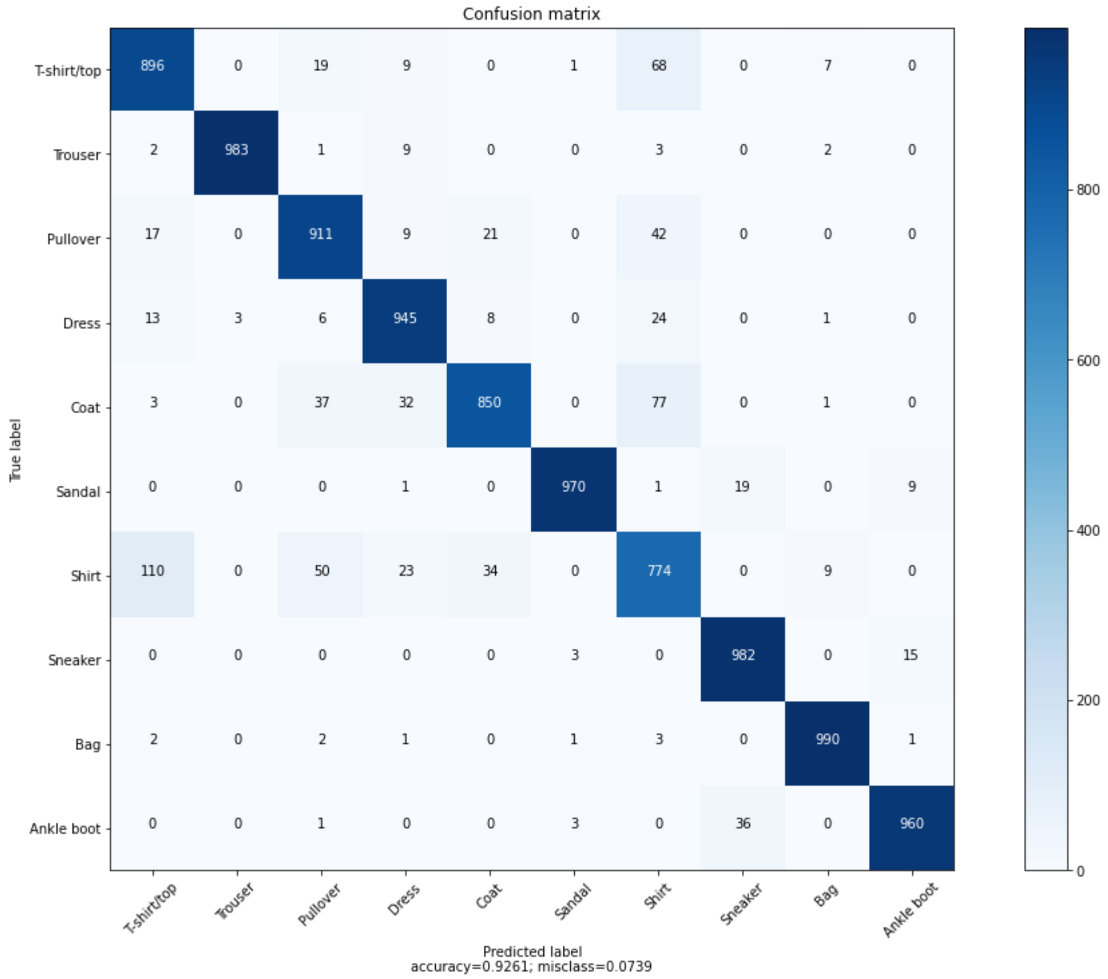

Figure 6. CNN confusion matrix

Discussion, Comments/Notes and Future Work

Even though our model exhibited strong results, it does not mean that there is no room for improvement. The dataset that was used covered a specific number of fashion product categories and was limited in size. The expansion of dataset's size and the increment of the number of classes could lead to a stronger model that can generalize in diverse situations. The future additional classes could relate not only to clothing types but also to wearable equipment like smartwatches and headphones. The later version of the model could be integrated by companies that do not operate only within fashion industry like shopping malls and hardware companies.

The classifier developed could also be used in conjunction with other artificial intelligence technologies. Its integration with an object detection algorithm could see many applications in different sectors. The algorithm we developed could also be used in conjunction with other artificial intelligence technologies. Imagine having an AI-powered classifier that stores could use to identify what people entering the store are wearing. This information could be used passively, to gather aggregate intelligence on what kinds of clothing retail customers typically wear. Or, it could be used actively. For example, a fashion detector could alert sales staff whenever a customer enters the store wearing a dress or a suit. Since this person is already wearing upscale, high-value clothing, they're more likely to spend more money in the store today.

Bibliography

1. Han Xiao, Kashif Rasul, Roland Vollgra, 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:1708.07747v2.
2. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03381v1.