

Machine Learning and Content Analytics

Author: Bellos Adam, f2822009

September 2022

Contents

Introduction.....	2
project, vision and goals.....	2
Data Collection and Dataset Overview.....	3
Algorithms, architectures/systems.....	4
Experiments – Setup,Configuration.....	6
Results & Quantitative Analysis	7
Discussion, Comments/Notes and Future Work.....	9
Bibliography/References.....	10

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[5].

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[6].

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 as 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

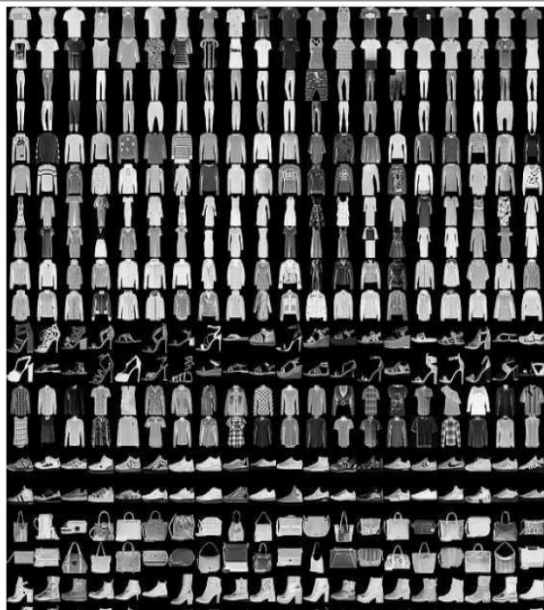
Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	

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 its 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.

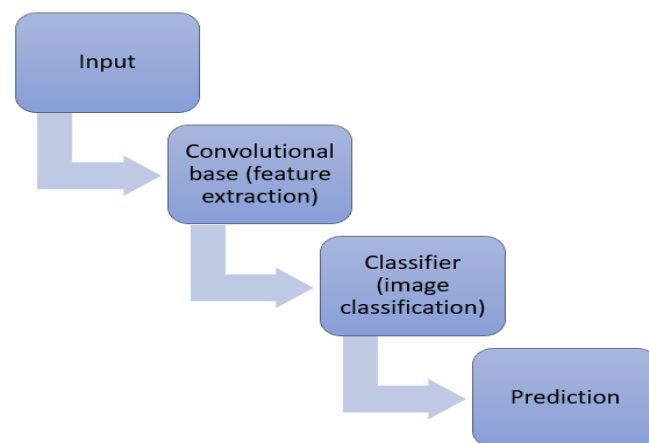


Figure 2. Parts of CNN

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 four models were implemented and compared. One CNN (fig 3) was built from scratch and three 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. Lastly, a pre-trained vision transformer was implemented [3].

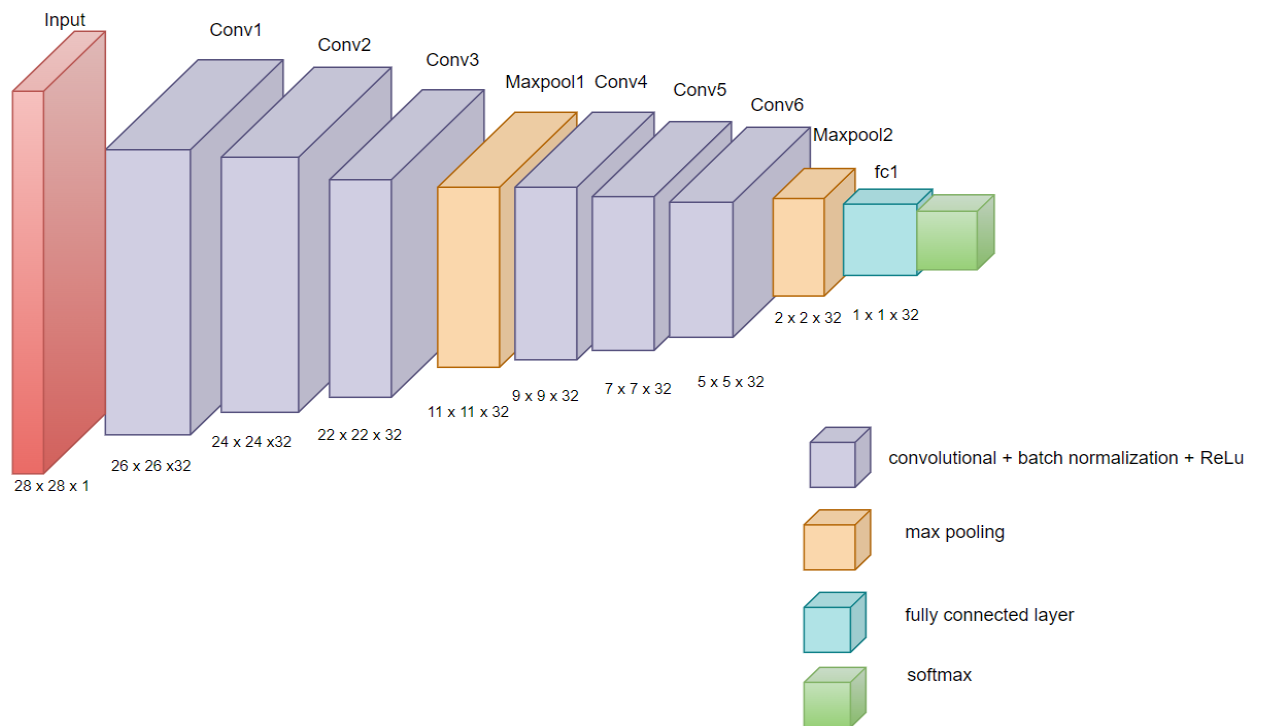


Figure 3. Architecture of the CNN.

Experiments – Setup, Configuration

CNN

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 4th, 6th and 8th step
- Gamma: 0.1
- Optimizer: Adam

Vision transformer (ViT)

The use of the transformer architecture is widely spread to natural language processing (nlp) tasks. The success of the architecture in the nlp field drove researchers to use transformers also for computer vision tasks. Initially transformers were used in conjunction with CNNs. Although, research has shown that a pure transformer applied directly to sequences of image patches can outperform the aforementioned models on image classification tasks when pre-trained on large amounts of data (14M-300M images) and transferred to multiple mid-sized or small image datasets. The main difference between the CNN and transformer architecture is that a transformer doesn't consider local attention as the CNNs do, but it considers global attention by taking advantage of its attention layer (in our case multi-head self-attention layers). Transformers are data-hungry, especially because they have the freedom to look everywhere onto the image from the start. In other words, initially a transformer is very unfocused at first and it needs huge amount of data to learn how to focus and what to focus its attention. With a huge amount of data the

transformer can find unexpected and unique ways to look the data because there is no element in the architecture that tells the models what and where to look. A CNN however is focused from the beginning by the convolutions to a local view. Additionally, CNNs run through the input sequentially while transformers do it in parallel. The pre-trained transformer model (fig 4) used for comparison is pretrained on ImageNet-21k dataset [4].

The parameters of the model implemented are as follows:

- Dimensionality of the encoder layers and the pooler layer: 768
- Number of hidden layers in the Transformer encoder: 12
- Number of attention heads for each attention layer in the Transformer encoder: 12
- Dimensionality of the intermediate feed-forward layer in the Transformer encoder: 3072
- The non-linear activation function in the encoder and pooler: gelu
- The dropout probability for all fully connected layers in the embeddings, encoder, and pooler: 0.0
- The dropout ratio for the attention probabilities: 0.0
- The standard deviation of the truncated_normal_initializer for initializing all weight matrices: 0.02
- The epsilon used by the layer normalization layers: $1e-12$
- The size (resolution) of each image: 224
- The size (resolution) of each patch: 16
- Factor to increase the spatial resolution by in the decoder head for masked image modeling: 16
- The number of input channels: 3
- bias to the queries, keys and values: True

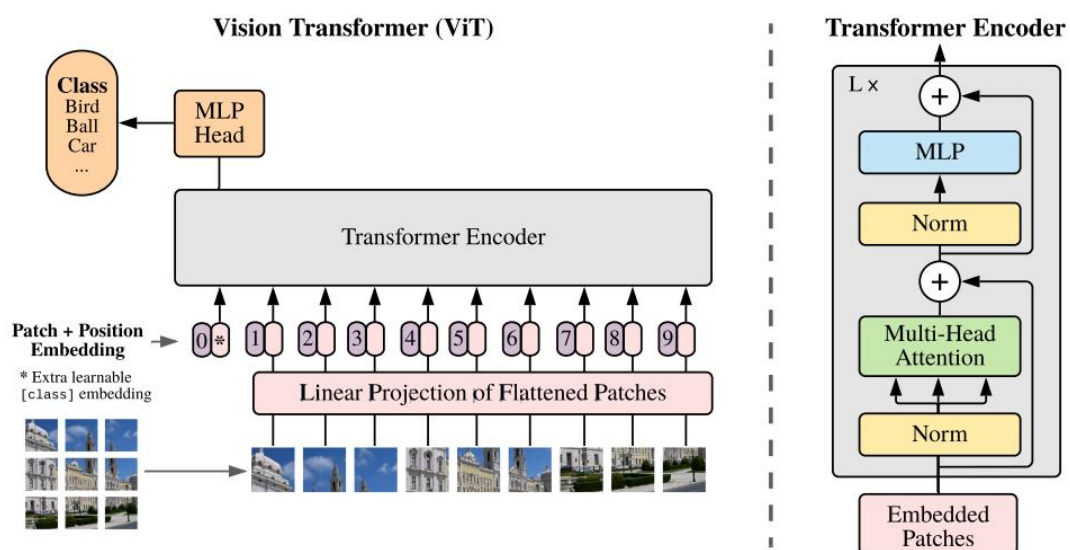


Figure 4 Vision transformer architecture

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, the ResNet50, the ResNet18, the pre-trained vision transformer, a 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 as they achieve higher scores. The models are compared based on their accuracy (fig5). Informally, accuracy is the fraction of predictions the model got right, meaning the number of images classified correctly. Formally, accuracy has the following definition:

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

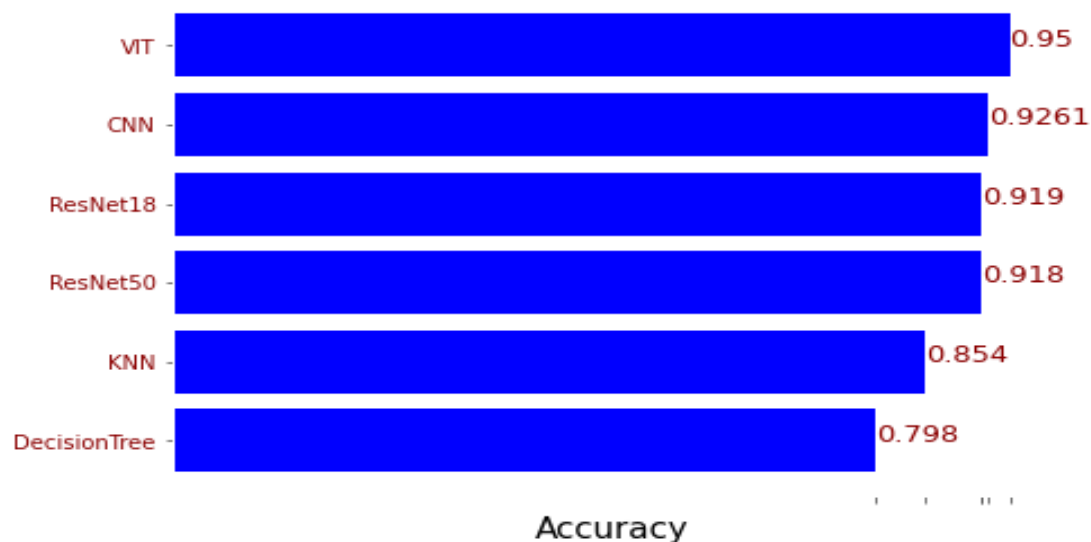


Figure 5. Model comparison in regard with accuracy

As it can be inferred deep learning models perform better compared to traditional models 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 6). 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.

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

	Model	Precision	F1_score	Recall
0	VIT	0.949980	0.949940	0.950000
1	CNN	0.926600	0.926000	0.926100
2	ResNet18	0.919000	0.919000	0.918300
3	ResNet50	0.918000	0.918000	0.918400

Figure 6 Recall, Precision and F1 scores of the deep learning models

From the model comparison we can see that ResNet18 and ResNet50 have similar performance while the CNN and VIT models seem to perform better. In particular, by looking at the F1 score which is the harmonic mean of precision and recall we can conclude that the vision transformer model outperforms all the deep learning models that were implemented. The vision transformer model may present the best results but is computationally more demanding in terms of execution time and of computational resources needed. Thus, when we face a similar classification task one should consider this trade off before selecting which model to implement. To get a deeper understanding of how well VIT and CNN perform we can see the fraction of correct predictions per class on fig 7 and fig 8. On the confusion matrices of fig 7 and 8, vertical axis represents the class to which the respective image belongs, while the horizontal axis represents the class to which the model has categorized the image. On the diagonal we see the number of images correctly classified by the model. As we have seen, the CNN model correctly classifies about 93% of the images and VIT 95% of them. Although, for both models several failures appear in classes that are similar among themselves, such as the shirt and T-shirt class, the shirt and coat class. As expected, the better performing VIT model misclassifies less images out of those similar items.

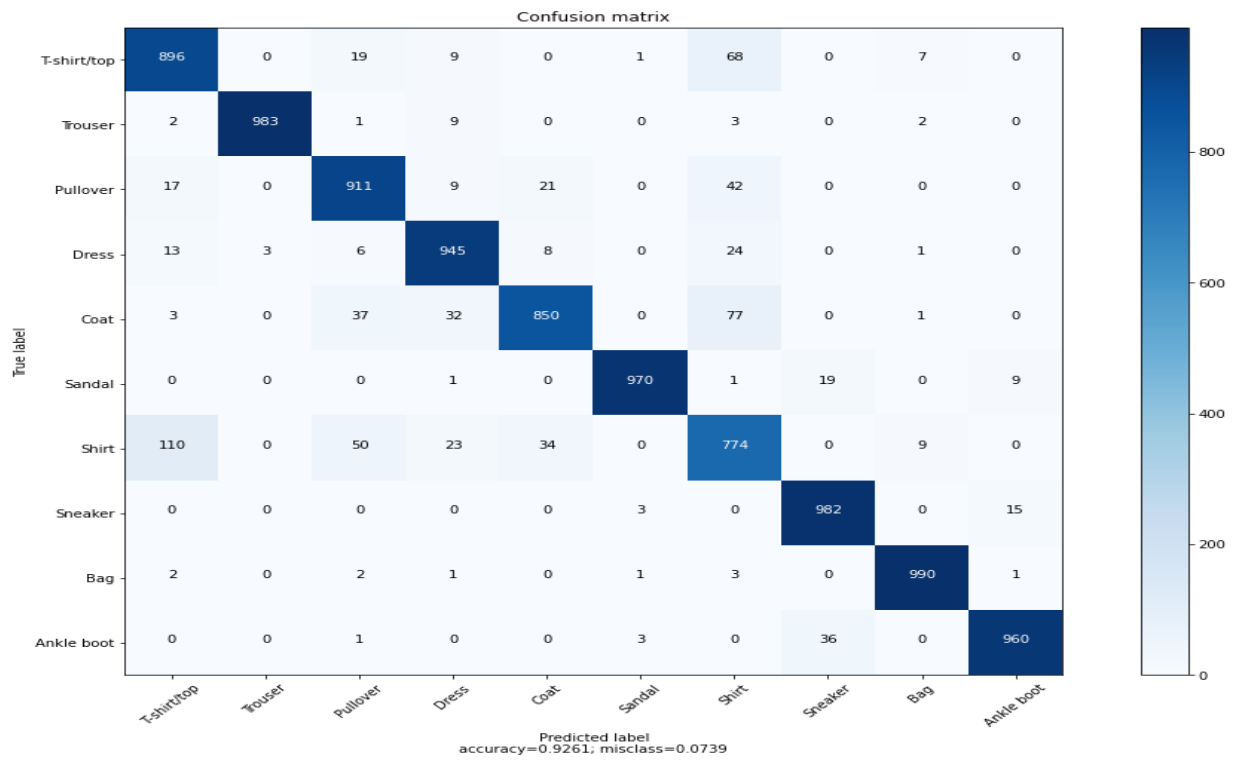


Figure 7. CNN confusion matrix

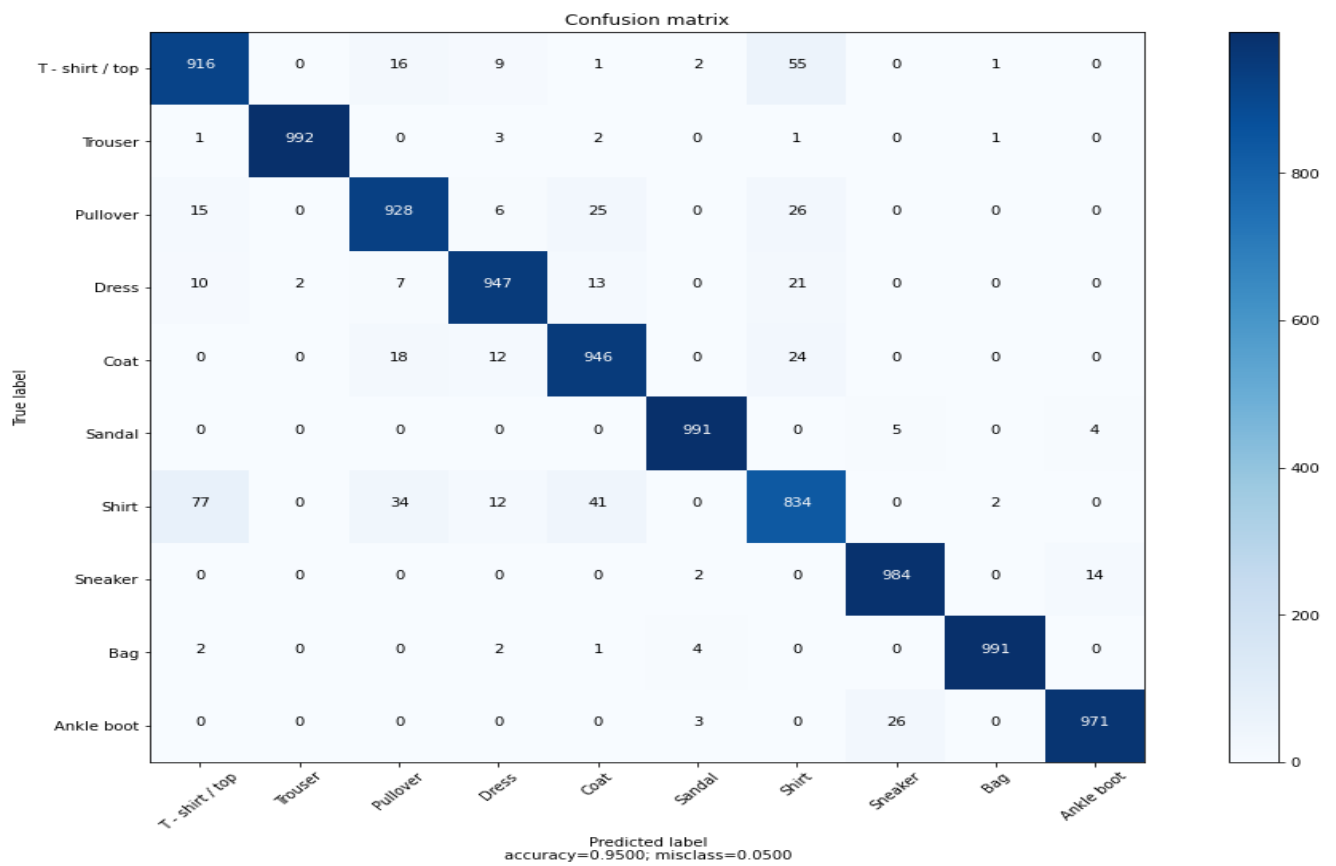


Figure 8 VIT confusion matrix

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. Another alternative is considered to be a more complex dataset where each image won't contain just one fashion item that needs to be classified, consider a set of images taken from social media platforms or fashion magazines. In this situation images would contain a background and most probably a set of non-fashion related items. 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 implemented models 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. 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/References

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.
3. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby, June 2021. AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE. arXiv:2010.11929
4. Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, Lihi Zelnik-Manor, August 2021. ImageNet-21K Pretraining for the Masses. arXiv:2104.10972
5. Zippia.com. 28 DAZZLING FASHION INDUSTRY STATISTICS [2022]. <https://www.zippia.com/advice/fashion-industry-statistics/>

6. Retaildive.com. Retail spending on AI to reach \$7.3B by 2022.
<https://www.retaildive.com/news/retail-spending-on-ai-to-reach-73b-by-2022/516170/>