Conquering Fashion MNIST With CNN Using CV

O.Pandithurai
Associate professor,
Department of computer science and Engineering
pandics@ritchennai.edu.in

G.saikrishnan
Department of mechanical engineering
Rajalakshmi institute of technology
saikrishnan.g@ritchenai.edu.in

S.vivek
department of mechanical engineering
Rajalakshmi institute of technology
vivek.s@ritchenai.edu.in

1.Padma Priya.A
Computer Science
Rajalakshmi Institute of Technology
Padmapriya.a.2021.cse@ritchennai.edu.in

2.Rithanya.R.K
Computer Science
Rajalakshmi Institute of Technonolgy
rithanya.r.k.2021.cse@ritchennai.edu.in

3.Pavithra.M
Computer Science
Rajalakshmi Institute of Technology
pavithra.m.2021.cse@ritchennai.edu.in

Abstract—This paper presents classification of garments from fashion MNIST using Deep learning concept Convolutional neural network (CNN) . Online fashion market is constantly growing, and an algorithm capable of identifying garments can help companies in the clothing sales sector to understand the profile of potential buyers and focus on sales targeting specific niches, as well as developing campaigns based on the taste of customers and improve user experience.

Keywords— Convolutional network, open CV, sequential modelling, preprocessing,tensorflow keras function.

I. Introduction

The Fashion MNIST dataset is a widely used benchmark in the field of computer vision and machine learning. It consists of a collection of 60,000 grayscale images of fashion items, divided into 10 different categories. These categories include t-shirts, trousers, pullovers, dresses, coats, sandals, shirts, sneakers, bags, and ankle boots.

In this project, we aim to develop a Convolutional Neural Network (CNN) model to classify these fashion items accurately. CNNs are a type of deep learning model that excel at processing and extracting meaningful features from images. They have revolutionized the field of computer vision and have been employed successfully in numerous applications, including image recognition, object detection, and face recognition.

By leveraging the power of CNNs, we will construct a model that learns to recognize the unique visual patterns and characteristics associated with each fashion category. The project will involve several steps, including data preprocessing, model architecture design, training, and evaluation.

Through this project, we will gain valuable insights into the application of deep learning in the fashion industry. The ability to automatically classify fashion items can have significant implications, from improving online shopping experiences to enhancing inventory management systems. Our goal is to build an accurate and robust CNN model that can serve as a foundation for various fashion-related applications.

II. Literature Survey

Fashion MNIST has attracted considerable attention in the field of computer vision and machine learning, leading to a wide range of literature exploring effective approaches for classifying fashion items using this dataset. One prominent paper, "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms" by Han Xiao, Kashif Rasul, and Roland Vollgraf (2017), introduces the Fashion MNIST dataset as a replacement for the traditional MNIST dataset. It evaluates various machine learning algorithms, such as support vector machines, decision trees, and neural networks, on this dataset. Another article, "Fashion-MNIST: A Step Towards Algorithmic Transparency in Machine Learning" by Laurence Moroney (2018), discusses the dataset's significance in promoting algorithmic transparency and suggests the use of alternative evaluation metrics beyond accuracy. "Fashion-MNIST: A Case Study on the Robustness of Transfer Learning" by Xiaojuan Qi, Qibin Hou, Jun Wan, and Xiaoou Tang (2018) investigates the performance of transfer learning techniques, comparing fine-tuning and feature extraction approaches using pre-trained models. Additionally, "Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units" by Shao-Hua Sun, Hua Zhang, Chi Zhang, Xiang Zhao, and Jian Sun (2019) explores the impact of activation functions on CNN models, introducing the Concatenated Rectified Linear Units (CReLU) activation function. Lastly, "Fashion-MNIST: A Versatile Benchmark for Image Recognition" by Hiroyuki Nakayama, Yuichiro Kanzawa, and Masayuki Tanaka (2020) provides a comprehensive overview of the dataset, discussing evaluation metrics, data augmentation techniques, and model architectures used by different researchers. These studies collectively contribute to our understanding of the challenges and advancements in fashion image classification using Fashion MNIST.

III. Objective

The objectives for working with the Fashion MNIST dataset can be summarized as follows:

1. Classification Accuracy: The primary objective is to develop a machine learning model, specifically a Convolutional Neural Network (CNN), that achieves high accuracy in classifying fashion items correctly. The aim is to build a model that can accurately identify the category of each item in the dataset.

2.Model Generalization: It is crucial to create a model that generalizes well to unseen data. The objective is to ensure that the trained model performs well not only on the training data but also on new, unseen examples. This is important for the model's practical applicability in real-world scenarios.

3. Comparative Analysis: Another objective is to compare different approaches, techniques, or architectures for fashion item classification using the Fashion MNIST dataset. This involves

evaluating the performance of various models, exploring the impact of different hyperparameters, and determining which methods are most effective for this specific task.

4. Interpretability and Explainability: Understanding the decision-making process of the model is important, especially in industries like fashion where interpretability is crucial. The objective is to explore techniques that provide insights into why the model made certain predictions. This could involve visualizing the learned features or generating explanations for the model's decisions.

5. Robustness and Resilience: Fashion MNIST models should exhibit robustness and resilience to variations in input data. The objective is to create a model that can handle different image qualities, lighting conditions, occlusions, and other potential challenges that may arise when classifying real-world fashion images.

6.Optimization and Efficiency: Optimizing the model's performance and efficiency is another objective. This includes exploring techniques such as model compression, quantization, and pruning to reduce the model's size and computational requirements while maintaining high accuracy.

7. Real-World Applications: The ultimate objective is to apply the developed model to real-world applications within the fashion industry. This could involve integrating the model into recommendation systems, inventory management tools, online shopping platforms, or other fashion-related applications.

By addressing these objectives, researchers and practitioners aim to advance the state of the art in fashion image classification and contribute to the development of practical and efficient solutions for various fashion-related tasks.

IV. Outcomes

The outcomes of working with the Fashion MNIST dataset can be summarized as follows:

1. High Accuracy Fashion Classification: One of the main outcomes is achieving a high level of accuracy in classifying fashion items. The developed machine learning models, particularly the Convolutional Neural Networks (CNNs), should demonstrate superior performance in accurately identifying and categorizing different types of fashion items.

2. Comparative Performance Analysis: The project aims to provide a comparative analysis of different approaches, techniques, or architectures for fashion item classification using the Fashion MNIST dataset. The outcomes will include insights into the strengths and weaknesses of various models and a better understanding of which methods yield the best performance.

3. Model Generalization: The trained models should exhibit good generalization capabilities, ensuring that they perform well on unseen data beyond the training set. This outcome is crucial for the practical applicability of the models in real-world scenarios, where they may encounter new and previously unseen fashion images.

4. Interpretability and Explainability: An important outcome is gaining insights into the decision-making process of the models. Techniques such as feature visualization and explanation generation can provide a better understanding of the learned representations and the factors influencing the model's predictions, thus enhancing interpretability and explainability.

5. Robustness and Resilience: The models should demonstrate robustness and resilience to various challenges commonly encountered in fashion image classification, such as variations in image quality, occlusions, and lighting conditions. The outcome is the development of models that can handle these variations effectively and provide reliable predictions.

6. Optimization and Efficiency: Another outcome is optimizing the models for efficiency without compromising performance. Techniques like model compression, quantization, and pruning can reduce the model's size and computational requirements, leading to more efficient inference and deployment in resource-constrained environments.

7. Real-World Applications: The ultimate outcome is the application of the developed models to real-world fashion-related applications. This could involve integrating the models into recommendation systems, inventory management tools, or online shopping platforms, where they can enhance the user experience, improve fashion item search and discovery, and provide valuable insights for business decision-making.

V. Challenges

One of the most common challenges in training a Convolutional Neural Network is overfitting. There are several ways to mitigate this problem in a Deep Learning model: increasing the number or size of layers, or use a technique known as dropout. The term dropout refers to “dropping” units (neurons) from a neural network, which means, temporarily removing them from model. The choice of which neuron to remove is random, and the amount can be fixed using a constant, for example, the constant 0.5 defines that half of the neurons will be removed. This technique has considerably improved the accuracy and performance of neural network in several applications, such as object classification, speech recognition and document classification, among others. This shows that this technique can be generalized for any problem.

VI. Architecture

1.Convolution Layer: The first to extract features from an input image is the convolution layer. The convolutional layer preserves the relationship between pixels by using only a tiny input square for image information. It is a two-step process: First, there needs to be an image matrix. Second, this image matrix is processed with a kernel or Filter . (i) The image matrix has a width, height, and depth of h ×w× d (ii) The Filter’s size is fh × fw× d (iii) The output’s length, width, and height are (hfh+1) × (w-fw+1) ×1 A 3x3 filter matrix is used to make a 5x5 image whose pixel values are 0, 1. The output of the convolution of a 5∗5 image matrix multiplied by a 3∗3 filter matrix is called “Features Map”. To blur a picture, apply a convolution and sharpen it, apply a convolution. And to detect edges, apply a convolution. Convolutions feature extraction benefits from strides and padding.

2.Strides: How the filter convolves around the input volume is determined by its stride. In the first scenario, the filter convolved around the input volume by moving one unit at a time. The stride refers to how much the filter shifts. Astride is commonly chosen to create an integer rather than a fractional volume. The stride is the number of pixels that are shifted across the input matrix. The filters are shifted one pixel at a time when the stride is set to 1, and two pixels at a time when the stride is set to 2. 3.1.2. Padding. Padding is a term used in convolutional neural networks to define how many pixels are added to an image by the CNN kernel when it processes it. If the padding in a CNN is set to 0, the value of every pixel added will be zero. Padding is essential in the construction of a convolutional neural network. If we shrink the image and use a

neural network with hundreds of layers, we will end up with a small image that has been filtered.

3.Pooling Layer: In image preprocessing, the pooling layer is very critical. Pooling reduces the number of parameters when the photos are too huge. The act of "downsizing" the picture generated by the preceding layers is known as "pooling." Reducing the density of an image by making it smaller is equivalent to reducing the image's pixel density. By reducing the number of distinct features of each map, spatial pooling, also known as downsampling or subsampling, reduces dimensionality without losing valued data.

Spatial pooling includes:

4.Max Pooling: Maximum pooling is a discretization procedure that samples the value. One of the critical functions of the binning sub-region minimization is to constrain assumptions about features discovered in the binned region. By applying a max filter to the sub-regions of the initial representation that are not overlapping, it is possible to obtainthe greatest possible pooling. Figure 6 shows the procedure of Max pooling.

5. Average Pooling: Average pooling is a down sampling process for feature maps that first determines the average value for patches of a feature map and then uses it to generate a down-sampled feature map. Once a convolutional layer is applied, it's commonly used. The input will be split into rectangular regions and the average values of each sector will be calculated. Calculating the average for each patch of the feature map is what average pooling entails. This means that each of the feature map's 2x2 squares gets down sampled to its average value.

6.Sum Pooling: Feature map down sampling, which reduces the pixel size of images, is made possible by the Sum Pooling. In other words, sum pooling is a form of a max function, but instead of returning the maximum value, it takes the sum of all the input values. Although they use the same sub-region as their max function, mean and sum pooling choose to use their sub-max regions instead of using the max function.

7.Fully Connected Layer: Every input from one layer is connected to each activation unit of the following layer, and the layers are completely coupled. A standard machine learning model almost always incorporates fully connected layers at the end that take in all of the information presented by earlier layers and ultimately deliver the final result. At the output of the convolutional layers, the data features are represented at a high level. You can use a fully connected layer to learn non-linear combinations of these features while keeping the output layer flattened and connected to the output layer. The fully connected layer gets input from other layers and converts it to a vector before sending it. The output will be divided into the desired number of classes by the network.

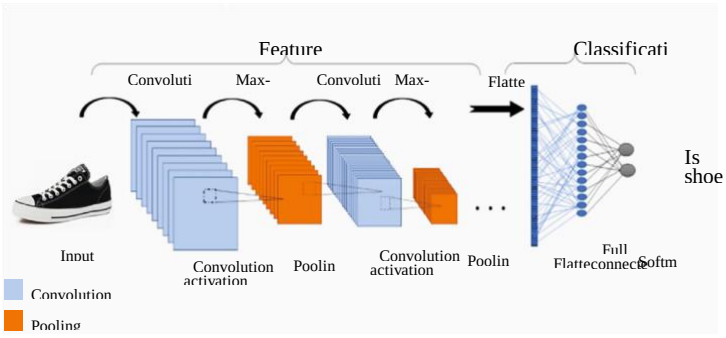
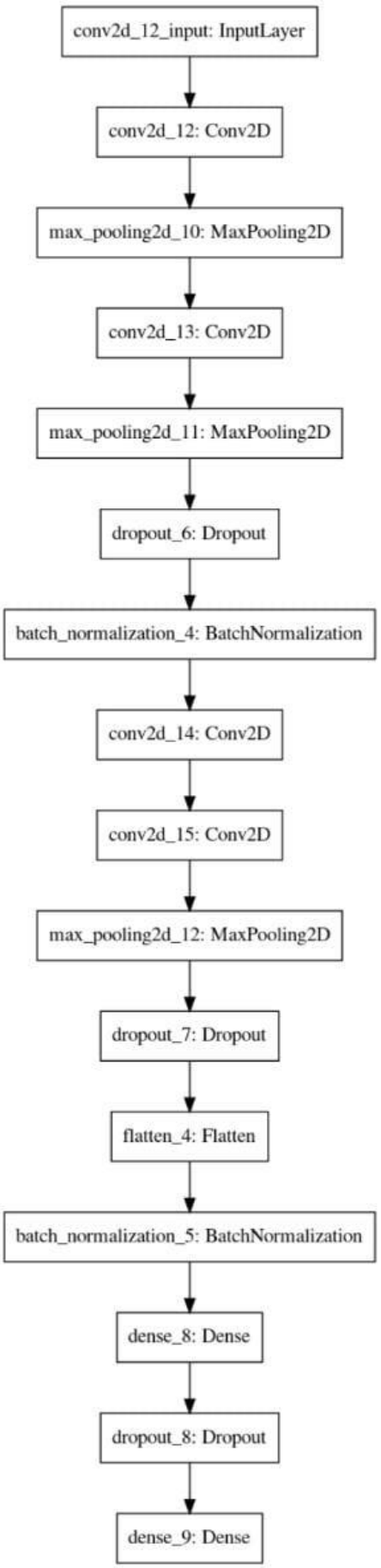
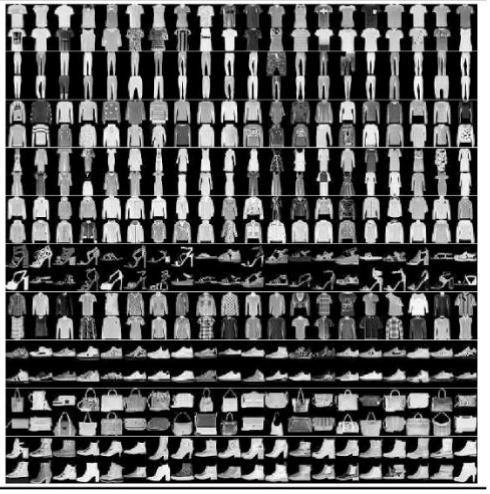


FIG: CNN for image classification.



VII. IMPLEMENTATION :

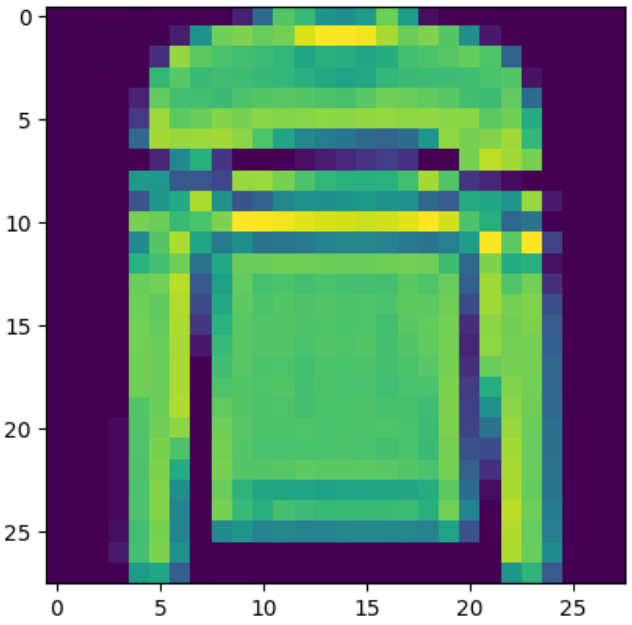
Table 2. Class names and example images in Fashion-MNIST dataset.

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	

Above mentioned is the dataset for our classification. which contains 70,000 grayscale images in 10 categories. The images show individual articles of clothing at low resolution (28 by 28 pixels)

1.PREPROCESSING :

The data must be preprocessed before training the network. If you inspect the first image in the training set, you will see that the pixel values fall in the range of 0 to 255

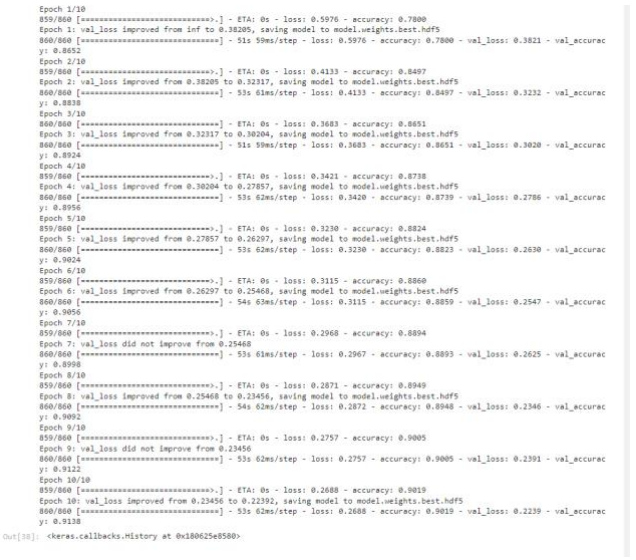


To label this dataset, four CNN models were done in Python with Keras and TensorFlow. Training was executed in a Jupyter notebook. We also used Weights and Biases [14] to grab information about training and hardware usage. Proposed models were named: cnn-dropout-1, cnn-dropout-2, cnn-dropout-3 and cnn-simple. The goal of those models was to be able to label the dataset without the need of too much training or processing on activation, so developers can use it on real time applications such as online stores and searching websites.

cnn-dropout-1 and cnn-dropout-3 Both models use two consecutive blocks containing: a convolution, a max pooling, and finally a drop out. These blocks are connected to two more fully connected layers, who are connected to an output layer of ten neurons, each one representing a category. The only difference between these two models is that cnn-dropout-3 has considerably lower dropout values, where the first drop out values is for model 1, and the second one for model 3. This topology has 44426 trainable parameters.

cnn-dropout-2 :This proposed model is very similar to the cnn-dropout-1 model. However, it has two layers of convolutions before each max pooling. This model has about 32340 trainable parameters

cnn-simple : Cnn-simple is a model with less layers. It has only two convolutions, followed by a fully connected layer, in addition to the respective dropout and max pooling like other models. This model has 110968 trainable parameters. Since this model has only one max pooling, the image gets to the dense layer with 14x14 pixels in size (four times the size of other models which is 7x7). So, de dense layer training is expected to be slower. All those models were modeled based on Keras Sequential model. Convolutional and dense layers used Rectified Linear Unit (ReLU) activation functions, except by the last dense layer on each model (output layer), were Softmax was used. The optimizer used was Adadelta [28] Batch size was 128 and we trained the models for 12 epochs. To improve results, image pixel luminosity values were normalized to float numbers between 0 and 1.



MATERIAL AND METHOD :

To Understand Better about the model, let us consider the Below Example. According to Figure 8, the handwritten image will be divided into convolutional filters for feature extraction; if the image is large, pixels filters will be used to divide it into different parts. In the above Fig., the image has been divided into three filters. After extracting features, it is done with Activation Functions and Pooling. The extracted features are then in 2D or 3D Arrays. With the help of the flatten layer, it should be converted into 1Dimensional Arrays, and the dense layer should

help one neuron connect to another with fully connected layers. This should classify the image. 4.1. Filters Formation. The Filters Formed wherever the object is identified it filled with some value. Filters can detect the patterns such as edges in an image by detecting the values of an image. A convolutional layer's filters typically learn to recognise concepts like a person's shoulders or face borders. We may achieve a greater level of abstraction and in-depth knowledge from a CNN by stacking additional layers of convolutions on top of each other. the computers transformed the image into Matrix Format. In the matrix, the cells filled with number 1 like wherever the object is identified, are filled with one remaining; all are -1. In this way, the image went divide into three filters which we discussed earlier. 4.2. Feature Map Formation. In Figure 10. Shows how the feature map extracted is shown. The matrix form of image and Filter is multiplied to get the feature map. Firstly, it gone started with the first cell and ended with the last cell.

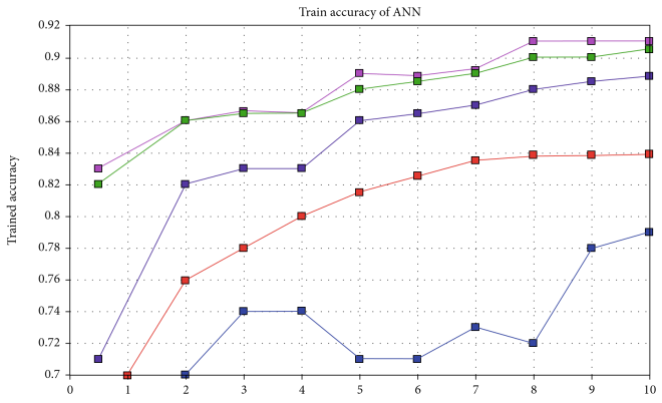


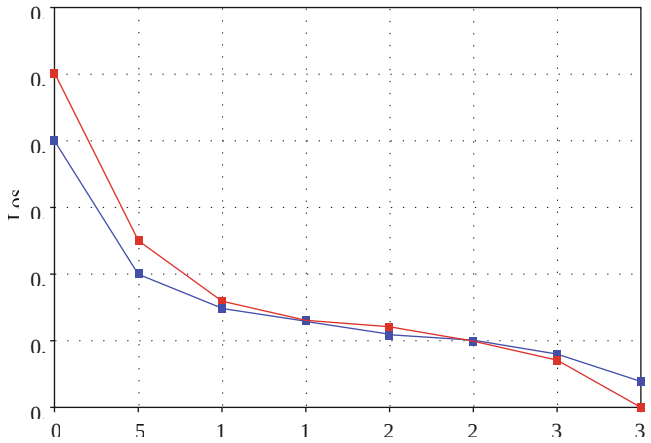
FIG:Comparing Best Optimzer with existing methods

ACCURACY:

```
test_loss, test_acc = model.evaluate(test_images,  test_labels, verbose=2)
print('\nTest accuracy:', test_acc)
```

```
313/313 - 1s - loss: 0.3446 - accuracy: 0.8803 - 597ms/epoch - 2ms/step
Test accuracy: 0.8802999985408783
```

LOSS FUNCTION:



VIII.RESULTS:

The classification accuracy for each class was plot using subplot.

IX.Conclusion:

In this paper, we address the problem of distinguishing clothing elements in fashion photographs using a CNN. This is accomplished using the fashion MNIST dataset, which contains many images, together with CNN and ANN algorithms for accurate and effective image classification. Image recognition is becoming increasingly crucial as deep learning algorithms progress. CNN recognition is commonly used when it comes to fashion-related applications, such as garment classification, retrieval, and computerised clothing labelling. In this study, we employ CNN architecture on the Fashion MNIST dataset. We would like to compare this with different datasets. Fashion MNIST is a dataset having low-resolution images. We would like to try these high resolution images in the future, and we also plan to put CNN architecture to the test on a dataset of real-life apparel pictures that we amassed ourselves.

X.ReferencePapers:

[1] A. Hodecker, A.M.R. Fernandes, A. Steffens, P. Crocker, V.R.Q. Leithardt, "Clothing Classification Using Convolutional Neural Networks," in Iberian Conference on Information Systems and Technologies, CISTI, IEEE Computer Society, 2020, doi:10.23919/CISTI49556.2020.9141035.

[2] R. Boardman, R. Parker-Strak, C.E. Henninger, "Fashion Buying and Merchandising," Fashion Buying and Merchandising, 2020, doi:10.4324/9780429462207.

[3] Y. Zhong, S. Mitra, "The role of fashion retail buyers in China and the buyer decision-making process," Journal of Fashion Marketing and Management, 24(4), 631–649, 2020, doi:10.1108/JFMM-03-2018-0033.

[4] K.V. Madhavi, R. Tamilkodi, K.J. Sudha, "An Innovative Method for Retrieving Relevant Images by Getting the Top-ranked Images First Using Interactive Genetic Algorithm," Procedia Computer Science, 79, 254–261, 2016, doi:10.1016/j.procs.2016.03.033.

[5] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, L. Van Gool, "Apparel classification with style," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 321–335, 2013, doi:10.1007/978-3-642-37447- 0_25.

[6] H. Chen, Z.J. Xu, Z.Q. Liu, S.C. Zhu, "Composite templates for cloth modeling and sketching," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1, 943–950, 2006, doi:10.1109/CVPR.2006.81.