Conquering Fashion MNIST with CNNs using Computer Vision

Jayakumar B  
Computer Science and Engineering  
Rajalakshmi Institute of TechnologyChennai, India  
[jayakumar.b.2021.cse@ritchennai.edu.in](about:blank)Desika C  
Computer Science and Engineering  
Rajalakshmi Institute of TechnologyChennai, India  
[desika.c.2021.cse@ritchennai.edu.in](about:blank)

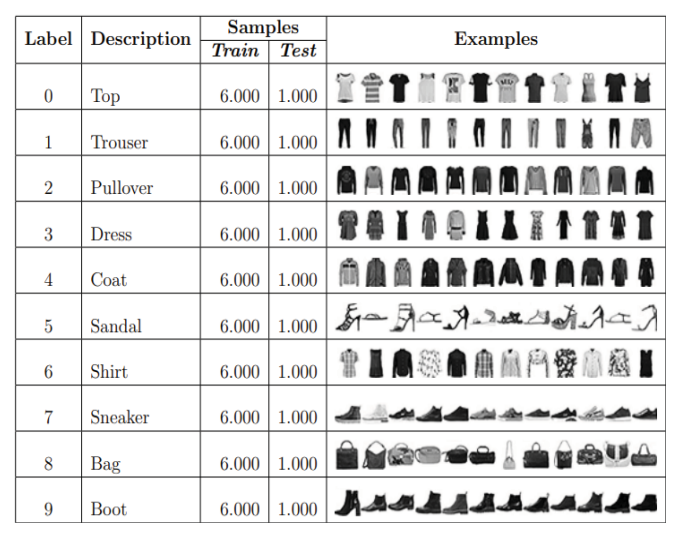
**Abstract**

***The online sales market was having the rapid growth and they were having the capable algorithm of identifying the fashion items clothes. This can help the sales section to understand the profile of customers and focus on the sales target. The artificial intelligence was approaching the consumer and make them understood the importance of cloths. That were used to improve the sales market convolutional neural networks were used in image classification this paper would represent the different type of convolutional neural network that were constantly used in Fashion MNIST. They help to classify the fashion items such as cloths. The efficiency of the image classification was shown by the convolutional neural network. To classify the products such as clothes to identify and help the researchers by the Fashion MNIST data set. The main goal of the project is to classify the product based on comparison. They have the accuracy of 98.83%***

***Keywords: convolutional neural network (CNN), computer vision (CV), Fashion-MNIST, Batch Normalization, hyperparameters.***

# Ⅰ. INTRODUCTION

In the recent 20-30 years the fashion industry has the rapid change, in the market. Increasing the profit by understanding the buyer’s taste. Through these website people can their clothes that result in the rise of online business. They help the customers when searching. When searching items in these platforms. The customer taste culture was also helping the labeling by the kind of information. The Europe’s largest online fashion platform is based on the fashion MINST data images. The 70000 products are divided into 10 categories with 28 \*28 pixel grey scale images from fashion MNIST.Those divided categories: shirt,bags,trousers,ankle boots coat,pullovers,sandles,dress,tshirt.In the artificial intelligence this dataset were widely used in web store ,those images were formaly thumb nail and it is uploaded in keras. For datalabling the different artificial intelligence models were work on this dataset which one better suited.The implementation were Decision tree,Support vector classification,Stochatic gradient descent ,Extra tree,Gaussian naïve bayes,Gradient boosting ,K-neighors,linear support vector classification,Logistic regression ,multilayer perceptron,passive aggressive classifier,perceptron and random forest.

The best result of these implementation can give the accuracy of 98.83%. 

# 2. LITERATURE SURVEY

## Classification of Fashion Article Images using CNN:

The photos of Zalando fashion items make up the Fashion MNIST dataset, which includes a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image. Each image is associated with a label from one of ten classifications. One of the most fundamental problems in computer vision is image categorization, which has several useful applications, including indexing images and videos. Although it is relatively easy for a human to recognize a visual object in an image, it is extremely difficult for a computer system to do the same with the accuracy of a human. Convolutional neural networks have achieved excellent results in image classification, image segmentation, computer vision, and natural language processing.

Inspired by neurobiology, convolutional neural networks generally involve three phases. In the first phase, we use a set of filters or kernels to build a feature map by overlaying them on the input image. These filters or kernels are often quite small — usually 3x3, 4x4, or 5x5. A dot product of the filter settings and the area of the image over which the kernel slides is added element by element as it moves over the image. The following convolution operation is performed on the maximum or average output of the summarized output if the mesh has many layers. An image is classified using a multilayer perceptron mesh after a series of convolution, activation, and pooling processes are completed.

## Application of CNN in Clothing Image Classification:

Although some traditional machine learning methods such as Support Vector Machine (SVM), Principal Component Analysis (PCA), and Linear Discriminant Analysis (LDA) have been used for image classification, these methods still have their own limitations when processing large amounts of image data. Convolutional neural networks (CNN), recurrent neural networks (RNN), and other deep neural networks are used in Deep Learning to address the lack of large-scale data processing power. Categorization of garments at a finer level is required for the desired classification results. According to the study cited in the article, lower-level CNN results in features such as distinct stains and forming edges in photos of dogs. Higher-level CNN, on the other hand, produces intricate features such as the fine details in dog photos. Hierarchical categorization can be used in this situation for the layers that make up CNN.

The hierarchical structure of clothing photos is classified using HCNN. The benchmark model is tested without hierarchical structure to allow comparison of model results. Two convolutional layers and a pooling layer form the first two building blocks of the HCNN model based on VGG16. The input of a coarse-grained first-level classifier is the output of these two building blocks. Three convolutional layers and a pooling layer are used to merge the third and fourth building components. The second stage coarse-grained classifier uses the output of these two building blocks as input. The output of the final fine-grained classifier is the input of the fifth building block, which contains three convolutional layers and one pooling layer. The projected values of coarse 1, coarse 2, and fine 3 are each determined after the input image passes through the HCNN model. For example, if we input an image of a jumper, the first coarse categorization branch is likely to represent clothing, the second is likely to represent a coat, and the last branch would likely predict a jumper as the output

.

# 3. OBJECTIVES

## Image Classification:

The main goal of the Fashion- MNIST CNN model is to categorize images. The model learns to recognize the many types of fashion products after being trained on a dataset of photographs of clothing items. After training, the model can

be used to categorize new images of garments.

## Feature Extraction

To extract features from photographs, a model can also be used. Other tasks, such as image retrieval, object identification, and image segmentation, can then be performed using these features. The features that the model extracts are generally high-level features that capture the most important elements of the image.

## Model Benchmarking:

The model can be used to compare the performance of different CNN architectures and hyperparameters. In this way, the optimal model architecture and hyperparameters for a given application can be found. The performance of the model can be evaluated against a test set that must not be changed.

## Education and Research:

The concept can be applied to both academic and scientific endeavors. For example, it can be used to teach students about CNNs and image categorization. It can also be used to study new CNN designs and hyperparameters. The model can create visualizations of the retrieved features. Understanding how the model generates its predictions can be supported by these graphics.

# 4.OUTCOMES

## Interpretability:

By displaying the features that the model of Fashion- MNIST CNN has identified, the model can be made understandable. This makes it easier to understand how the model makes its predictions and to find out the key elements of an image that the machine considers when classifying it.

## Style Recommendations:

Based on customers' previous buying behavior and the photos they liked, a fashion- MNIST CNN model could be used to suggest styles to them. This could be used to improve the user experience in fashion e-commerce applications.

## Fashion Forecasting:

A Fashion-MNIST to predict fashion It is possible to predict fashion trends using the CNN model. Retailers could use this to stock their shelves accordingly, and fashion manufacturers could use it to design their product ranges.

# 5. CHALLENGES

In machine learning, generalization refers to how successfully the learned concepts of the model are applied to examples not seen during training. The goal of most ML models is to generalize well from training data to later make good predictions for unknown (test) data. Overfitting occurs when models adapt too well to small details and noise from training data, but do not generalize well, resulting in poor performance on test data. Regularization is an important part of avoiding overfitting. The following are some of the regularization strategies used in this work to avoid overfitting:

## Dropout:

In the dropout method, random neurons are ignored or dropped during training. When neurons are randomly removed from the network, the remaining neurons must mediate and manage the necessary representation to make predictions for the missing neurons. Normally, one assumes that the network learns many representations as a result. As a result, the network is less sensitive to specific neuron weights. As a result, the network gradually becomes better equipped to deal with speculative situations and less inclined to over-adjust the training data. Dropout results in a model that is more trustworthy and resistant to early overfitting.

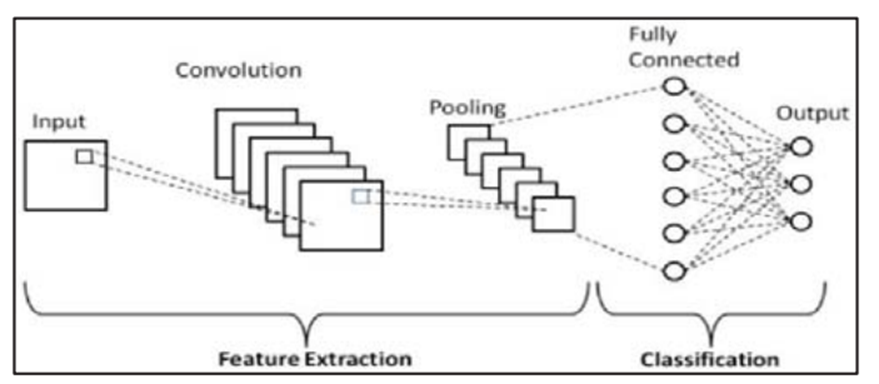
## Early Termination:

Stopping the training process early is another strategy to avoid overfitting. Since the model usually only gets worse after a certain number of epochs if it continues to be trained, the training should be stopped as soon as the validation loss increases. In this study, early stopping is used to increase the training speed of the model (2-layer and 3-layer CNN).

## Batch Normalization:

Batch normalization reduces covariance shift, the amount by which hidden unit values move. By subtracting the batch mean and dividing it by the standard deviation of the batch, BN normalizes the yields of a past activation shift to increase the stability of an NN and accelerate learning. After each CONV layer in the work reported in this study, batch normalization was used to accelerate the training of the model (3-layer CNN).

# 6. ARCHITECTURE MODEL



The input for the first layer (C1) is a 3232-grayscale image. This image is processed by the first convolutional layer, which contains six feature maps or filters with a size of 55 and a step size of 1. The image is reduced from 32x32x1 to 28x28x6 pixels.

Second layer (S2): This pooling layer has six feature maps, each of which is 14 by 14 pixels. Each unit of a feature card is connected to a 2 x 2 block in the corresponding feature card in C1. S2 contains 5880 connections and an additional 12 trainable features. The final image is resized to 14x14x6.

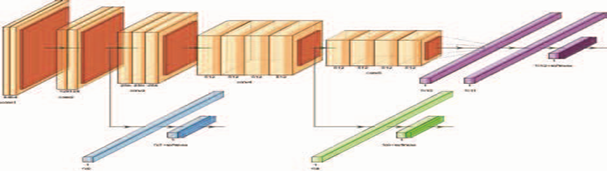
The third layer (C3) consists of 16 convolutional layers, each with 5 by 5 features and a stride of 1. The units of each feature map relate to different 55 blocks corresponding to points in a subset of the feature maps of S2. In the last layer, the data of all feature maps of S2 are loaded.

Fourth layer (S4): This is also an average pooling layer with a 22-unit filter and a 22-unit stride, and 16 feature maps, each 55 in size. Each unit of a feature map is connected to a 2 x 2 block in the corresponding feature map in C3, following the same path as C1 and S2. Apart from 32 trainable parameters and 2000 links for output, this layer is reduced to 5x5x16, corresponding to the second layer (S2).

Fully connected convolutional layers with 120 feature mappings form the fifth layer (C5). On each of the 16 feature maps in S4, a 5-block corresponds to a particular unit. Since the feature map dimension would be greater than 11 if the LeNet 5 input were made larger while everything else remained protected, C5 is categorized as a convolutional layer rather than a fully connected layer. All four hundred nodes (5x5x16) inside the fourth layer, S4, are related to every of the 120 units within the fifth layer, C5.

Sixth layer (F6): A layer of 84 units that is fully connected. It is fully connected to C5. 10164 trainable parameters are present.

7. SOFTWARE MODEL



HCNN Model Based on VGG16 Framework

A variety of variables, including shift parameters, hyperparameters, and NETWORK ARCHITECTURE, can be changed. The implementation procedure started with a simple single-layer CNN model (with ReLU, no padding, and dropout) and then expanded the layers and units to achieve the optimal validation performance while solving the overfitting problem.

20 epochs of a simple 1-layer CNN model developed using the following architecture:

(1) A 32-by-3x3 filter CONV with ReLU and no padding

(2) TWO X 2 pooling in a max-pooling layer

(3) Dropout.

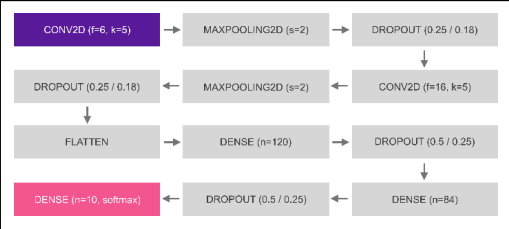
(4) Flatten layer

(5) The 128-neuron Fully Connected Layer (ReLU)

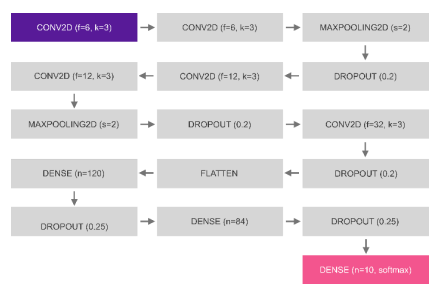
(6) Dropout 10 neurons form the output LAYER. The seventh layer of the CNN implements the SoftMax function with categorical cross-entropy (to calculate the loss).

8. CNN MODELS

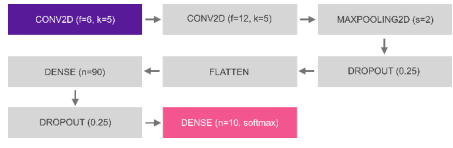
To label this dataset, four CNN models were created in Python using Keras and TensorFlow. Training was performed in a Jupyter notebook using GPU. We also used Weights and Biases [14] to obtain information about training and hardware usage. The proposed models were named: cnn-dropout-1, cnn-dropout-2, cnn-dropout-3 and cnn-simple. The goal of these models was to label the dataset without requiring too much training or processing in activation so that developers can use them in real-time applications such as online stores and search websites.

8.1. cnn-dropout-1 and cnn-dropout-3 Both models use two consecutive blocks that contain a convolution, a max-pooling, and finally a dropout. These blocks are connected to two more fully connected layers, which in turn are connected to an output layer of ten neurons, each representing a category. The only difference between these two models is that cnn-dropout-3 has significantly 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.

8.2. cnn-dropout-2 This proposed model and the cnn-dropout-1 have some similarity.Before each max-pooling, it has two layers of convolutions. This model which has about 32340 trainable parameters

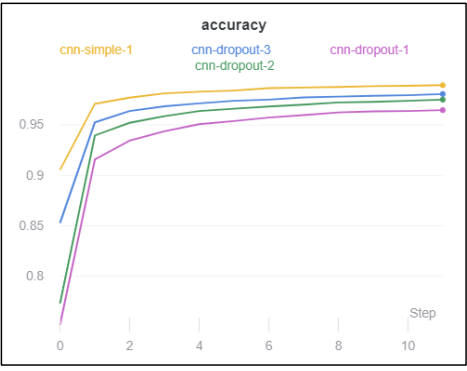
8.3. cnn-simple Cnn-simple is a model with fewer layers. It has only two convolutional layers followed by a fully linked layer, in addition to the respective dropout and max-pooling as in other models. This model has 110968 trainable parameters.



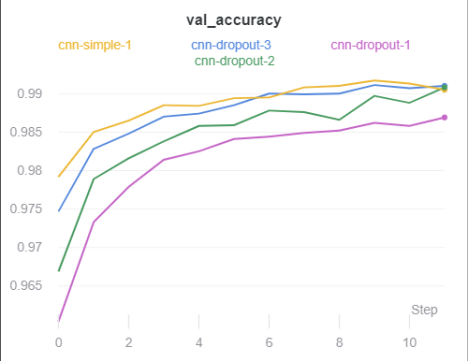
Since this model has only maximum pooling, the image reaches the dense layer with a size of 14x14 pixels (four times larger than the other models, which are 7x7). Thus, training of the dense layer is expected to be slower. All of these models were modeled based on the Keras Sequential model. Convolutional layers and dense layers used Rectified Linear Unit (ReLU) activation functions, except for the last dense layer of each model (output layer) where Softmax was used. Adadelta [28] was used as the optimizer. The stack size was 128 and we trained the models for 12 epochs. To improve the results, the brightness values of the image pixels were normalized to floating point numbers between 0 and 1.

9. RESULT

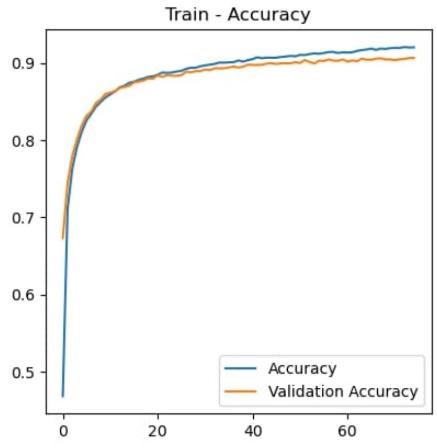
On the training data set, the most accurate model was CNN-simple with 98.83% accuracy. Other models were also acceptable based on the results obtained: 98.06% (cnn-dropout-3), 97.51% (cnn-dropout-2), and even the worst model (cnn-dropout-1) achieved an accuracy of 96.46%



There were mixed results for validation accuracy, with cnn-dropout-3 scoring the best at 99.1%, followed closely by cnn-dropout-2 (99.08%) and cnn-simple (99.05%). Cnn-dropout-1 still achieved the worst results (98.69%).



Loss had good results, with cnn-simple scoring the best in both training and validation loss, followed by cnn-dropout-3, cnn-dropout-2, and cnn-dropout-1. In the evaluation of time, the two more accurate models were also faster in training. Figure 10 shows the relationship between the time and the training epoch. The slope angle of the lines shows how fast the training was.



# 

# 

# 10. conclusion

Mode- MNIST CNN models need to be updated to keep pace with the growing size and complexity of mode data sets. This may require the use of larger and more complex CNN architectures and more advanced training methods. In addition, fashion- MNIST CNN models will need to improve their interpretability. Developers will then be able to understand how the models generate their predictions and identify the critical elements of an image that the algorithms use for categorization. In general, Fashion- MNIST CNN models are effective tools for feature extraction and image categorization. They have several potential uses, and it is possible that they'll grow in importance over time.

11. References

### S. Bhatnagar, D. Ghosal, and M. H. Kolekar, "Classification of fashion article images using convolutional neural networks," 2017 Fourth International Conference on Image Information Processing (ICIIP), Shimla, India, 2017, pp. 1-6, doi: 10.1109/ICIIP.2017.8313740.

### Y. Liu, G. Luo, and F. Dong, "Convolutional Network Model using Hierarchical Prediction and its Application in Clothing Image Classification," 2019 3rd International Conference on Data Science and Business Analytics (ICDSBA), Istanbul, Turkey, 2019, pp.157-160, doi: 10.1109/ICDSBA48748.2019.00041.

### W. Di, "Comparative research on clothing image classification based on neural network models," 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT, Weihai, China, 2020, pp. 495-499, doi: 10.1109/ICCASIT50869.2020.9368530.

### L. Deng, “The MNIST database of handwritten digit images for machine learning research,” IEEE Signal Processing Magazine, 29(6), 141–142, 2012, doi:10.1109/MSP.2012.2211477.

### E. M. Dogo, O. J. Afolabi, N. I. Nwulu, B. Twala and C. O. Aigbavboa, "A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks," in Proceedings of 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), Belgaum, India, 2018, 92-99, doi: 10.1109/CTEMS.2018.8769211.