Image Classification Using deep learning Convolutional Neural Network

Victor. O. Owino

*owinohvicky@gmail.com*

Dr. Lillie Siele

10th June 2020

Table of Contents

[**Abstract** 3](#_Toc42637133)

[**Introduction** 4](#_Toc42637134)

[**1.1** **Statement of the problem** 4](#_Toc42637135)

[**1.2** **Research objective** 5](#_Toc42637136)

[**1.3** **Research questions** 5](#_Toc42637137)

[**1.4** **Overview of proposed convolutional neural networks** 5](#_Toc42637138)

[**Methodology** 6](#_Toc42637139)

[**2.1 Overall architecture** 6](#_Toc42637140)

[**2.2 Convolutional neural networks** 8](#_Toc42637141)

[**2.3 Convolutional layer** 9](#_Toc42637142)

[**2.4 Pooling layer** 10](#_Toc42637143)

[**2.5 Fully connected layer** 10](#_Toc42637144)

[**Simulation** 11](#_Toc42637145)

[**3.1** **Fashion MNIST dataset** 11](#_Toc42637146)

[**3.2** **Implementation** 14](#_Toc42637147)

[**3.2.1** **Exploring the dataset** 14](#_Toc42637148)

[**3.2.2** **Visualizing the dataset** 15](#_Toc42637149)

[**3.2.3** **Data preparation** 16](#_Toc42637150)

[**3.2.4** **Training the convolutional neural model** 17](#_Toc42637151)

[**4.0 Results and discussion** 19](#_Toc42637152)

[**5.0 Conclusion** 21](#_Toc42637153)

[**References** 22](#_Toc42637154)

# **Abstract**

There's been much speculation in recent years about neural networks technologies and other deep learning algorithms, primarily because of the popularity of several implementations in the sector utilizing these techniques. Consequently, this hype has yielded several innovative ideas to build open-source libraries and methods to enable the average income tech-savvies to achieve their objective. This research paper aims to examine and illustrate how to use deep learning technologies and algorithms to precisely classify a dataset of fashion images into their respective clothing categories. First, the paper shows the general knowledge of convolutional neural networks (CNN) and the techniques of image classification. Later on, it also discusses the methodology of building a neural network and the simulation process. The results of the neural network simulation are compressively evaluated and discussed.

Image Classification Using deep learning Convolutional Neural Network

# **Introduction**

Machine learning (ML) and artificial intelligence (AI) are techniques that have been in existence for quite a while. In recent days, ML and AI are more often applied in various sectors to improve performance and increase production. Advancements in the field of AI has indicated the rising development of Artificial neural network (ANN) and convectional neural networks (CNN) that that resembles the biological neural network. These developing techniques can perform better than the initial artificial intelligence and machine learning models. The convolutional neural network is an essential topic in these days, mainly when designing a deep learning system that handles images as its input. CNN has a simple neural architecture yet accurate in image processing and recognition (Keiron et, al., 2015). This paper provides a detailed discussion of deep learning CNN in image classification using the Fashion MNIST dataset. The models are based on Keras and TensorFlow frameworks. Following this chapter is the convolutional neural network methodology for this research, the CNN simulation process, and a comprehensive discussion of the achievable results of CNN in image classification.

## **Statement of the problem**

Fashion MNIST dataset consists of seventy thousand images divided int0 80% and 20% for the training and test set, respectively. This dataset consists of 28 x28 grayscale photos from ten different classes of fashion clothes. The types are a T-shirt/top, trousers, pullover, dress, coat, sandal, shirt, sneakers, bag, and ankle boot. The images are stored in the form of pixels with values ranging from zero and 255. There is a need for Machine learning engineers to use the image pixels and the description (label) encoding to categorize the images accordingly with the highest precision and accuracy percentage.

## **Research objective**

* This research aims to accurately and precisely classify the fashion clothing's images into their respective classes with an accuracy of 90% and above.
* To benchmark the best deep learning algorithms in image processing and classification.

## **Research questions**

* What class does each image belong to when accurately classified?
* Can deep learning neural networks classify image datasets precisely than the random classification algorithms?
* Does the layer of the neural network affect the accuracy of the prediction model?

## **Overview of proposed convolutional neural networks**

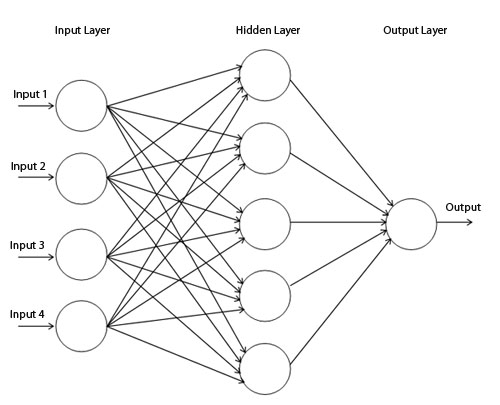
Image classification is a machine learning problem that ML experts to classify image using their pixel detentions and labels categorically. Often, humans understand models in the form of a visual impression while the computer understands images in the way of the pixel. The machine is only able to see the binary number from any model of an object, let's say orange fruit image. In human interpretation, the image can be represented in the decimal numbers from zero to 255. There are essential terminologies used in deep learning image classification. According to Nair (2018), the following are the critical concern while building a neural network:

1. *Convolution:* refers to the operation on an image dataset to identify specific features of that image. It helps in image blurring, sharpening, edge detection, and noise reduction to enable the machine to learn essential image features.
2. *Pooling:* often, raw images can become broad in size. Thus, pooling helps to reduce the image dimensions to avoid losing essential chrematistics.
3. *Flattening:* Neural networks uses a vector of features to learn and classify images. The purpose of flattening is to transform the 2D matrix of the image features into a vector quantity of building a neural network.
4. *Full connection:* refers to the procedure of feeding the flattened image features into the neural network.

# **Methodology**

## **2.1 Overall architecture**

Artificial neural networks (ANN) are algorithms of mathematical learning motivated by the properties of the genetic neural networks. They are used for a wide variety of purposes, ranging from reasonably necessary clustering topics to sophisticated voice recognition and computer vision applications. ANN is closely modelled on biological neurons in the context of being introduced as a series of correlations components, also known as nodes, that are functionally similar to human neuron connection. The ties between the numerous neurons (nodes) have arithmetic values (weights), and by consistently altering these values, the network will gradually estimate the desired functionality (Sootla, 2015).

Every node in the ANN network receives multiple inputs from other nodes and determines a single output dependent on the weights of the data and links. This output is usually fed into another neuron, and recurring the same procedure. If the knowledge provided in the previously trained model is fitted, it is simple to imagine the internal layer architecture of the artificial neural network. ANN architecture is ordered in three layers. The input layer receives the data, the middle layer, which is hidden, and the output layer that produce the network output. the figure below represents the three layers of ANN

*Figure 2: three-layered artificial neural networks*

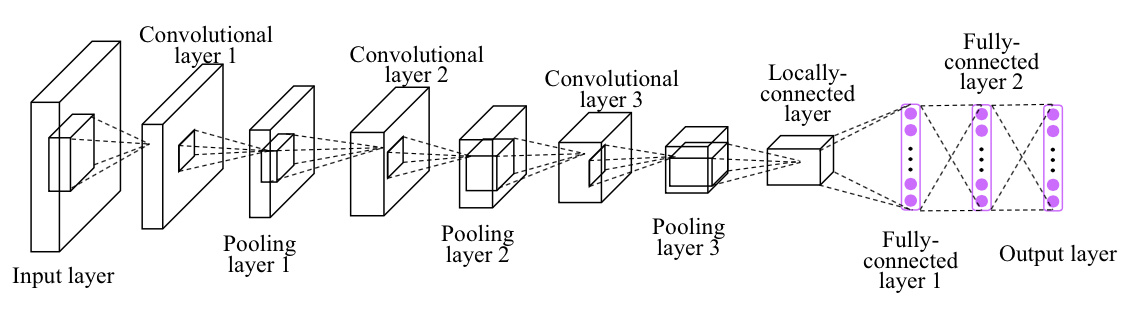
The concealed layers (hidden) may be viewed as discrete indicators of characteristics, growing awareness of intricate data patterns, as propagated over the network. For instance, if the system is designed to recognize a facial image, the first hidden layer could serve as a pixel detector. The second secret takes those pixels as their feedback and brings them together to shape a cloth. The third hidden layer matches the previous results to the exact class of clothing.

According to O' shea (2015), the two standard learning algorithms for image recognition and classification are supervised and unsupervised learning. Supervised machine learning is designed on the basis of pre-labelled data as its input. It accepts a vector input and produces a related output. The machine learns from experience using the labelled input and produces accurate prediction results through training and learning. On the contrary, unsupervised learning uses data without labels as its input, processes the input, and trains the machine to identify the unlabelled object through feature engineering. Often, image classification problems achieve the best accuracy with the labelled dataset.

## **2.2 Convolutional neural networks**

A convolutional neural network is the convectional ANN that is very effective and efficient in performing image recognition and classification tasks. CNN resembles the human way of visual recognition. It receives pixels as image inputs, transforms them using mathematical computations and activation function then builds a classifier for the input classes. CNN has an encode-specific feature in its architecture that suits it to perform image processing and computer vision tasks (O'shea and Nash, 2015).

CNN is applicable in areas such as image recognition, video analysis, natural language processing (NLP), drug discovery, and risk analysis in health. CNN architecture consists of three essential layers, namely, the convolutional layer, the pooling layer, and the fully-connected layer. Using the MNIST classification, CNN architecture can be represented as follows:

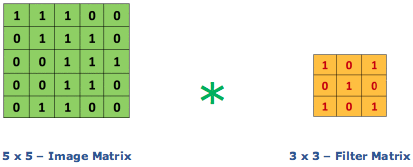


*Figure 3: Representation of CNN architectural layers*

The architectural layers and functionalities are explained as follows.

## **2.3 Convolutional layer**

Convolution layer is the first and a significant segment of CNN architectures that focuses on using kernels or filter parameters of image input. The kernels are small and spread dimensionally to the depth of the input parameters. Immediately the input parameters reach the convolutional layer, the layer it spreads each pixel across the input dimensionality and maps them into two directional maps using the activation function. For instance, consider an image of a 5 x 5-pixel dimension, the convolution layer multiplies image matrix with filter matrix (3 x 3) that is termed **as feature map and produces an** output represented below.



The convolution process is significant in achieving edge detection of the image, blur, and sharpens the image by convolving them with the filter matrix. The activation function (map) applies to corresponding filters, and the process is recurring.

The output of the convolutional layer is often optimized to using depth, stride, and padding parameters to avoid complexity. Stride refers to the number of image pixels that convolve over the input matrix. The depth of the output corresponds to the number of neurons within the convolutional layer. Reducing the depth parameter can reduce the total neuron complexity, which can also reduce the accuracy of the model to precisely recognize an image. Furthermore, padding provides effective control of the output dimensionality by padding the pixel edges (O'shea and Nash, 2015).

## **2.4 Pooling layer**

This is the second layer of CNN architecture that aims to minimize the parameters in cases where the image pixels are extremely large for the model. As a result, the computation complexity of the model is also reduced. This layer accepts the activation map of the preceding layer as its input parameter and scales it. This is called the max pooling, a type of spatial pooling that uses the max function. Other types of pooling that are applicable are the sum and average pooling, which reduces the dimensionality of the activation map while maintaining the essential features (O's hea and Nash, 2015).

## **2.5 Fully connected layer**

A fully connected layer is the last layer of the CNN architecture that flattens the output of the pooling matrix and feed it to the network of neurons. The neurons are interconnected and form the artificial neural network. In this layer, there are SoftMax and sigmoid activation functions that that summarises the entire tax of classifying the image. In summary, the image classification using convolutional neural network is a feed-forward process that following process (Prabhu, 2018):

* Initially, feed the image dimensions (pixels) into the convolutional layer.
* Correctly select the image features (parameters), apply strides and padding if it essential, and convolve the image using an activation map.
* Apply pooling to minimize the image dimensionality.
* Add more convolutional layers until the model achieves the best accuracy desired.
* Also, it flattens the results and feeds into the last layer called a fully connected layer.
* Finally, produce the result using an activation function, which classifies the target image into the respective class.

# **Simulation**

This chapter describes the implementation of a feed-forward convolution neural network with a single hidden layer capable of classifying the fashion clothing images into ten distinct groups using the Fashion MNIST dataset. The neural network is designed in python 3 using TensorFlow and Keras deep learning libraries. Other python libraries like pandas, NumPy, seaborn, and matplotlib are also used in data preparation and visualization. The code for the convolutional neural network is available at the Jupyter notebook attached to this file.

## **Fashion MNIST dataset**

Fashion MNIST is a simplified classification and computer vision dataset that is used by object recognition algorithms. It is a labelled subset of 70 thousand images in the dimension of 28x28 pixels collected by Zalando's article images. The data set is randomly divided into two, i.e., 60,000 training images and 10,000 test images. Each image is a 28x28 grayscale image that can be categorized into ten associated classes of fashion clothing. Zalando plans to act as a simple drop-in substitute for the initial MNIST dataset evaluating algorithms for machine learning and deep learning. This uses the same picture scale and arrangement with breaks for training and test dataset.

Here are the ten classes and some pictures that belong to the fashion MNIST dataset

| **Label** | **Description** |
| --- | --- |
| 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: Example of fashion MNIST dataset*

All pictures are 28 pixels in height and 28 pixels in width, totalling 784 pixels. Increasing pixel is correlated with a lower resolution-value, meaning the elegance or brightness of the image with higher numbers signifying brighter. This bitmap-value is several 0 to 255. There are 785 columns to the training and test data sets. The first column is comprised of class marks, which reflects the garment item. The remainder of the columns includes the corresponding picture pixel-values. Suppose we have decomposed x as x = i \* 28 + j, where i and j are figures between zero and twenty-seven, to detect a pixel on the image; the pixel is in row i and column j of matrix 28 x 28 (Zalando, 2017).

## **Implementation**

### **Exploring the dataset**

**Importing libraries**: As mentioned earlier, this convolutional neural network is designed using python 3, Keras, and TensorFlow. The design was done in the Jupyter notebook in anaconda environment. Initially, the dependent frameworks and libraries must be installed into a local machine then imported into the Jupyter notebook.

**Extracting dataset:** Initially, the fashion MNIST dataset is retrieved from the Kaggle website and stored in a local repository. The data set consists of train and test data. There are several ways of loading the data into the notebook, like using sci-learn or pandas. This research experiment used the Pandas library to load all the datasets and store them in the NumPy data frame.

**Features:** There are two essential features of the fashion MNIST dataset. They include image labels and pixels that represent the image dimensions. Often, human beings use their eyes to see, process, and distinguish object images. There exist receptor cells in the human eye that process the object pixels and represent it as an image. Likewise, the computer (machine) understands the photos as a two-dimensional array of integers (2D pixels). A machine can use either grayscale or RGB scale to understand the image boundaries of an object.

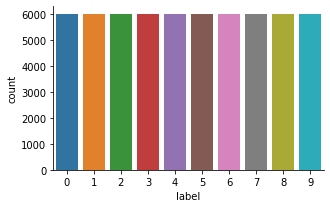
**Examine dimensions**: There are 60,000 training sets of images and 10,000 test image samples. All the images are represented 28 x 28 grayscale images that correspond to the clothing labels of the dataset. The pixels are integer values between zero and 255. The labels are encoded in integer values from zero to nine, each representing different classes of clothing, as shown in the dataset description.

**Check for null values:** This is an essential data cleaning process that entails checking for Null values in the dataset. Null values could be as a result of input errors and could cause anomalies in the model. Fortunately, fashion MNIST dataset does not have any null values.

### **Visualizing the dataset**

**Plotting random images:** It is essential to look deep into what pixels represent. This can be achieved by defining the classes and matching them with the pixels using random function.

**Distribution of description:** This is just to assess whether all the class labels are evenly distributed to avoid oversampling. The output of the distribution is showing that all the classes are evenly distributed.



### **Data preparation**

**Setting random seed: the** random seed is essential for reproductivity**.**

**Splitting data**: It is essential to have separate train and test data to avoid overfitting and underfitting errors. Scikit learn provides the simplest method to split the data. The train is used for training; validation is used to validate the accuracy of the model.

**Reshaping the image dimensions:** Keras deep learning library accepts an extra-dimensional object as its channel input. However, the pixels are representing in a 1-dimensional vector of 28 x 28 (748 pixels). The dataset is reshaped into (-1, 28,28,1), where -1 represents the first dimension, which is not known, and 1 represents the extra dimension for Keras channel.

**Normalization:** Normalization is essential to have a faster optimization algorithm. The pixel values are 8-byte integers that need to be scaled to [0,1] to achieve the optimization goal.

### **Training the convolutional neural model**

**Building the model**: This is the core where the neural convolution model is built. The following steps show the procedure of building a CNN model.

1. Initially, a sequential Keras API is defined to add the architecture layers one after another.
2. Then, adding the first CNN layer (convolutional layer), which is an essential building block of the CNN architecture. The target hyperparameters like filter numbers and dimensions (F), stride (S), padding (P), and activation function are manually input into the model. The depth of the output must be equals to the input number of filters. Mathematically, this is represented as follows:
3. *(Height, Width) = ((W-F +2P)/S) +1*
4. Next is adding the polling layer to reduce the dimensions of the sampling input. The pooling layer minimizes the parameters and computational power, thus reduce the rate of overfitting the model.
5. Then, batch normalization is added to reduce outliers and force the model to learn evenly without relying on specific weights.
6. Also, it is essential to have a dropout that realigns the model weight by dropping some neurons randomly.
7. And finally, the model is flattened and by adding the flattening layer and the output layer to classify the images using the SoftMax activation function.

**Compiling the model:** To calculate the accuracy of the model, the model must be compiled first. Compiling the model includes specifying the optimizer, loss function, and the metric evaluation method. The optimizer used in this research experiment is Adam. Also, this neural network experiment uses binary\_crossentropy for binary classification and categorical\_crossentropy for multi-class classification. Finally, the model uses accuracy to evaluate the model metrics.

**Model summary:** This is the stage of checking the review of the model to confirm that everything is as expected. The figure below shows the model summary.

|  |
| --- |
| Model: "sequential"  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Layer (type) Output Shape Param #  =================================================================  conv2d (Conv2D) (None, 28, 28, 64) 320  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  max\_pooling2d (MaxPooling2D) (None, 14, 14, 64) 0  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  dropout (Dropout) (None, 14, 14, 64) 0  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  conv2d\_1 (Conv2D) (None, 14, 14, 32) 8224  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  max\_pooling2d\_1 (MaxPooling2 (None, 7, 7, 32) 0  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  dropout\_1 (Dropout) (None, 7, 7, 32) 0  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  flatten (Flatten) (None, 1568) 0  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  dense (Dense) (None, 256) 401664  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  dropout\_2 (Dropout) (None, 256) 0  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  dense\_1 (Dense) (None, 10) 2570  =================================================================  Total params: 412,778  Trainable params: 412,778  Non-trainable params: 0 |

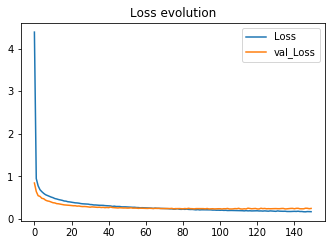
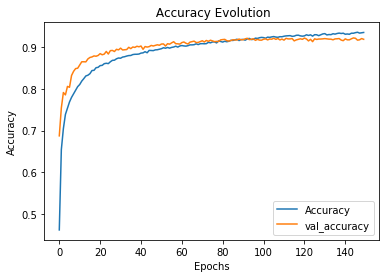
**Evaluating the model:** Plotting a confusion matrix summarise the model performance on the test data.

**Prediction and classification report:** Classification report visually represent the precision, recall, F1 scores, and support.

* Precision refers to the ration of true positive to the sum of a true and false positive. It represents the percentage of correctly predicted values.
* The recall represents the ability of a classifier model to identify all the possible true predictions. It's the ratio of true positive to the sum of true positive and false negative values.
* F1 score represents the aggregated mean value of precision and recall. A model with an F1 score of 1.0 has the highest accuracy level while that with zero ha the worse accuracy.
* Support refers to the actual occurrence of a class within a defined dataset. Imbalance support shows the weakness of the classifier model and can be corrected by stratifying the sampling.

# **4.0 Results and discussion**

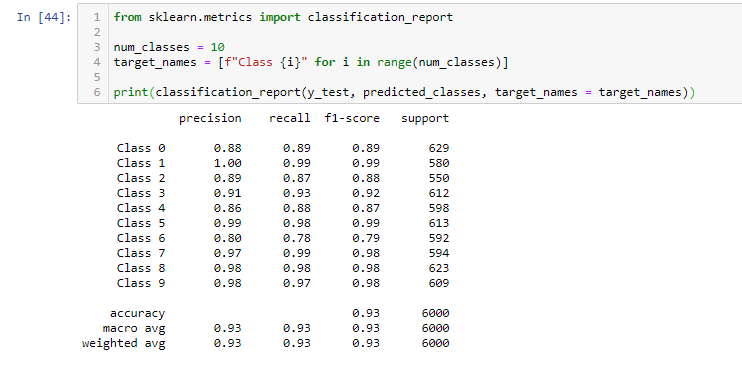
The success of the model was determined by the percentage of appropriate predictions provided by CNN during the study. A correct prediction means that the network classifies an image as the class to which it rightfully belongs. The outcome was evaluated for each class of the fashion clothes, and a mean result was determined for the entire subset. The training accuracy is 94%, while the accuracy on the validation set is 93%. A pot of loss evolution and accuracy evolution of the model is represented below. Loss value means how effectively a given model performs during each optimization iteration.



The graphs point out that the accuracy of the validation dataset (93%) is marginally less than that of the testing dataset (94%). The difference between the consistency of the preparation and the consistency of the test is a result of overfitting. Overfitting occurs when a model of machine learning performs poorly on new inputs that were previously unknown than on the training results. The loss of the CNN model is a negative lagging graph. This indicates that the models behave as expected with a reducing loss after each epoch.

Typically, the accuracy of a model is calculated after the parameters of the model are known and set, and no learning occurs. Then the test samples are fed to the machine, and after comparison with the true goals, the number of errors (zero-one loss) the machine produces is registered. Then, it measures the amount of misclassification.

The precision, recall, and F1 score for the model are fairly good. Class five of the fashion clothing is the most accurately predicted with a 99% f1 score.



# **5.0 Conclusion**

The convolutional model performs best in image classification with an accuracy of 93 %, which is above the hypothetical accuracy. However, there is an overfitting problem in which the model memorizes the training examples and becomes ineffective for the test set, causing the prediction accuracy slightly lower than the training accuracy. A creative data regeneration and backpropagation can correct the imbalance of the perdition accuracy.

# **References**

Nair, A. (2018). Learn image classification using CNN in Keras with code. *Analytic Indiaman*. Available at https://analyticsindiamag.com/learn-image-classification-using-cnn-in-keras-with-code/

Zalando Research. (2017). Fashion MNIST dataset. *Kaggle.* Available at https://www.kaggle.com/zalando-research/fashionmnist

Zalando Research. (2017). Fashion clothing categories and description. GitHub. Available at https://github.com/zalandoresearch/fashion-mnist/blob/master/doc/img/fashion-mnist-sprite.png

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Yang, S. Luo, P. Loy, C. K. W. Shum, X. (2015). Deep Representation Learning with Target Coding. CNN architecture. *Tang in Proceedings of AAAI Conference on Artificial Intelligence*, 2015

Prabhu. (2018). Understanding of Convolutional Neural Network (CNN) — Deep Learning. *Towards data science*. Available at https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148