# CONQUERING FASHION MNIST USING CNN AND COMPUTER VISION


**TEAM NAME:** The High Tide

**TEAM MEMBER'S**

K ROHIT REDDY

A SAMBA SIVA REDDY

K VAMSHI

**COLLEGE MENTOR:** Mrs D KALPANA

**INTEL MENTOR:** MOHAN IKAM

# ABSTRACT

This report focuses on the performance of several classifiers on the Fashion-MNIST dataset. Fashion-MNIST is a more difficult version of the classic MNIST benchmark image dataset. This dataset consists of 10 classes of 28x28 grayscale images of fashion items.

The data is normalized and principal component analysis and K-FOLD Algorithm are applied. K-nearest neighbors, and decision trees are evaluated on the dataset. Additionally, 5-layer back propagation neural networks and a convolutional neural network (CNN) are also tested.

Performance for these classifiers is compared using the data's built-in train/test split and using 5-fold cross validation. Additionally, k-means and winner-takes-all clustering techniques are investigated for visualizing and reproducing the clusters in the data. The CNN classifier achieves the best result of 91.3%.

## INTRODUCTION

The original MNIST dataset from 1998 is a popular 10 category dataset consisting of 70,000 examples of handwritten digits. It was first introduced in by LeCun et al. MNIST is a modified version of handwritten data obtained from NIST, the National Institute of Standards and Technology. The black and white images from NIST were normalized into a 20x20 pixel box, which preserved their aspect ratio. This resulted in grayscale images due to the anti-aliasing technique used in normalizing the images.

Finally, the images were centered in a 28x28 image based on the mass of the pixels. Originally created in 1998, MNIST has become an extremely popular benchmark dataset for its ease of use in prototyping and testing new classifiers. It is particularly popular in deep learning due to it small size compared to other datasets . In 2012, a record high accuracy of 99.77% was achieved using deep neural networks. Fashion-MNIST was introduced in 2017 by Xiao et al . In order to provide a similarly sized alternative to MNIST which poses a more challenging classification challenge.

To enhance the performance of classification models on the Fashion MNIST dataset, various techniques can be employed. These include data augmentation, where additional training examples are generated by applying transformations such as rotations, translations, and scaling to the original images. Additionally, model regularization techniques like dropout and weight decay can be used to prevent Overfitting.

Overall, the classification of the Fashion MNIST dataset presents an interesting and challenging problem in the field of computer vision. It provides a platform for developing and evaluating robust and accurate image classification models, paving the way for advancements in fashion-related applications and further research in the domain of computer vision and deep learning.

## BACKGROUND AND RELATED WORK:

Fashion-MNIST is a dataset of Zalando's fashion article images consisting of 60,000 training examples and 10,000 test examples.
Each example is a 28x28 grayscale image, associated with a label from 10 classes:
0: T-shirt/top
1: Trouser
2: Pullover
3: Dress
4: Coat
5: Sandal
6: Shirt
7: Sneaker
8: Bag
9: Ankle boot

Albawi et al. (2017) state that Convolutional neural network (CNN) is one of the most popular deep neural networks. O'Shea & Nash (2015) said convolutional neural network (CNN) is similar to artificial neural network (ANN) which is comprised of neurons that self-optimize through learning. CNN has multiple layers namely convolutional layer, pooling layer and fully connected layer. The main objective of convolutional layer is to obtain the features of an image by sliding smaller matrix (kernel or filter) over the entire image and generate the feature maps.

The pooling layer used to retain the most important aspect by reducing the feature maps. Fully connected layer interconnect every neuron in the layer to the neurons from the previous and next layer, to take the matrix inputs from the previous layers and flatten it to pass on to the output layer, which will make the prediction. Due to such architecture, it will take fewer parameter to learn and reduce the amount of data required to train the model.

## METHODOLOGY

K-fold cross-validation is a more reliable way to evaluate a model's performance It is a technique for evaluating the performance of a machine learning model on unseen data than simply training the model on the entire dataset and testing it on a hold-out test set.

It works by dividing the dataset into K folds, and then training the model on K-1 folds and testing it on the remaining fold. This process is repeated K times, and the average of the K test scores is used as the final estimate of the model's performance.

This is because k-fold cross-validation ensures that the model is not overfitting to the training data. To use k-fold cross-validation with Fashion-MNIST, we would first divide the dataset into K folds.

Then, we would train the model on K-1 folds and test it on the remaining fold. This process would be repeated K times, and the average of the K test scores would be used as the final estimate of the model's performance.

For example, if we were to use k-fold cross-validation with K=10, we would divide the dataset into 10 folds.
Then, we would train the model on 9 of the folds and test it on the remaining fold. This process would be repeated 10 times, and the average of the 10 test scores would be used as the final estimate of the model's performance.

K-fold cross-validation is a powerful technique for evaluating the performance of machine learning models. It is particularly useful for evaluating models that are prone to overfitting, such as deep learning models.

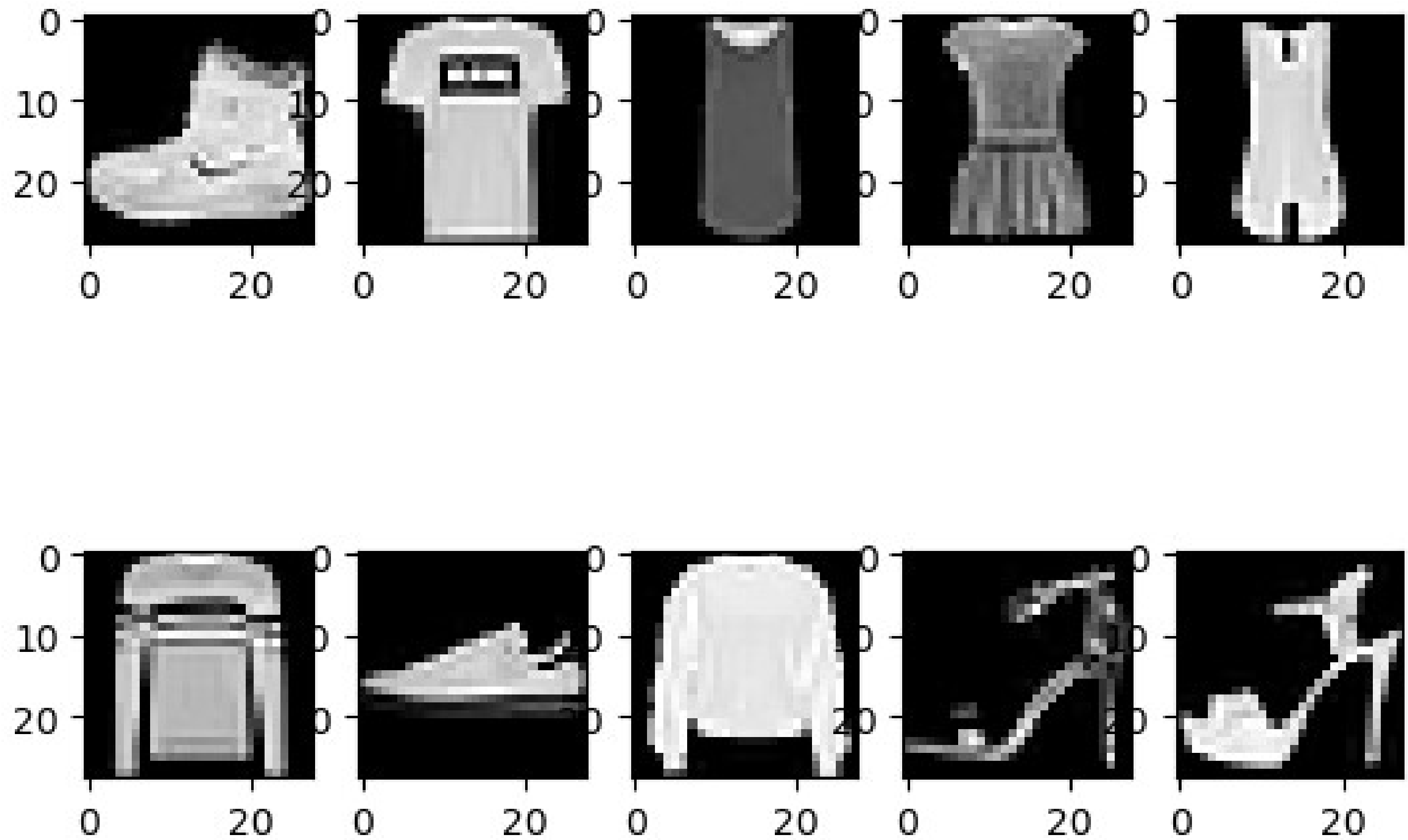
Here are some of the benefits of using k-fold cross-validation with Fashion-MNIST:

It provides a more reliable estimate of the model's performance than simply training the model on the entire dataset and testing it on a hold-out test set.

It helps to prevent overfitting by training the model on multiple subsets of the data. It can be used to tune the hyperparameters of the model to improve its performance.

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
    1. Take the group as a hold out or test data set
    2. Take the remaining groups as a training data set
    3. Fit a model on the training set and evaluate it on the test set
    4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores




**Building Model**

*A. Image Preprocessing*

The Fashion-MNIST database contains 70000 images of dimension 28x28. These images and their corresponding labels are separated as training data and test data. To prepare the data for training, some processing have applied on the images like resizing images, normalizing the pixel values etc. After doing the necessary processing on the image information's, the label data, we have converted it into categorical formats like label '5' should be represented as a vector format to build the model.

Among various deep learning architectures, ConvNets stands out for its unprecedented performance on computer vision. ConvNet is an Artificial Neural Network inspired by biological visual cortex and been successfully applied to image processing tasks. A special kind of artificial neural network is ConvNet which contains at least one convolutional layer. A typical ConvNet takes an input image, pass it through a set of layers convolution, non-linear activation, pooling (downsampling) and fully connected, and retrieve an output of classification labels. This output of this CNN layer is an activation map.

***1. Optimizers:***

Optimization algorithms help us to minimize or maximize the objective function. Minimizing the loss by the training process is very important and has a main role in the operation of training of the neural network model. The two optimizers used in these architectures are Adam [9] and Adadelta [20] for optimization of the loss function. Adam work well across a wide range of deep learning architectures. Adam usually outperforms the rest followed very closely by the other adaptive learning rate methods, Adagrad and Adadelta. Adam optimizer can be calculated as

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Adagrad is another popular gradient descent technique for optimization of loss function which is also used for the model parameters in our model. Adadelta prevents learning rate decay and it is an extension of Adagrad.

***2. Batch size and Number of Epochs***

Mini-batch is usually preferable in the learning process of ConvNets. A range of 16 to 128 batch size is a good choice to test with. ConvNet is sensitive to batch size. In this model we have used 64 and 128 as batch size for training images. Number of epochs is the number of complete pass through the entire training set. The number of epochs has increased until the difference of training and the test error is very small. Here, we have checked with 40 and 60 epochs.

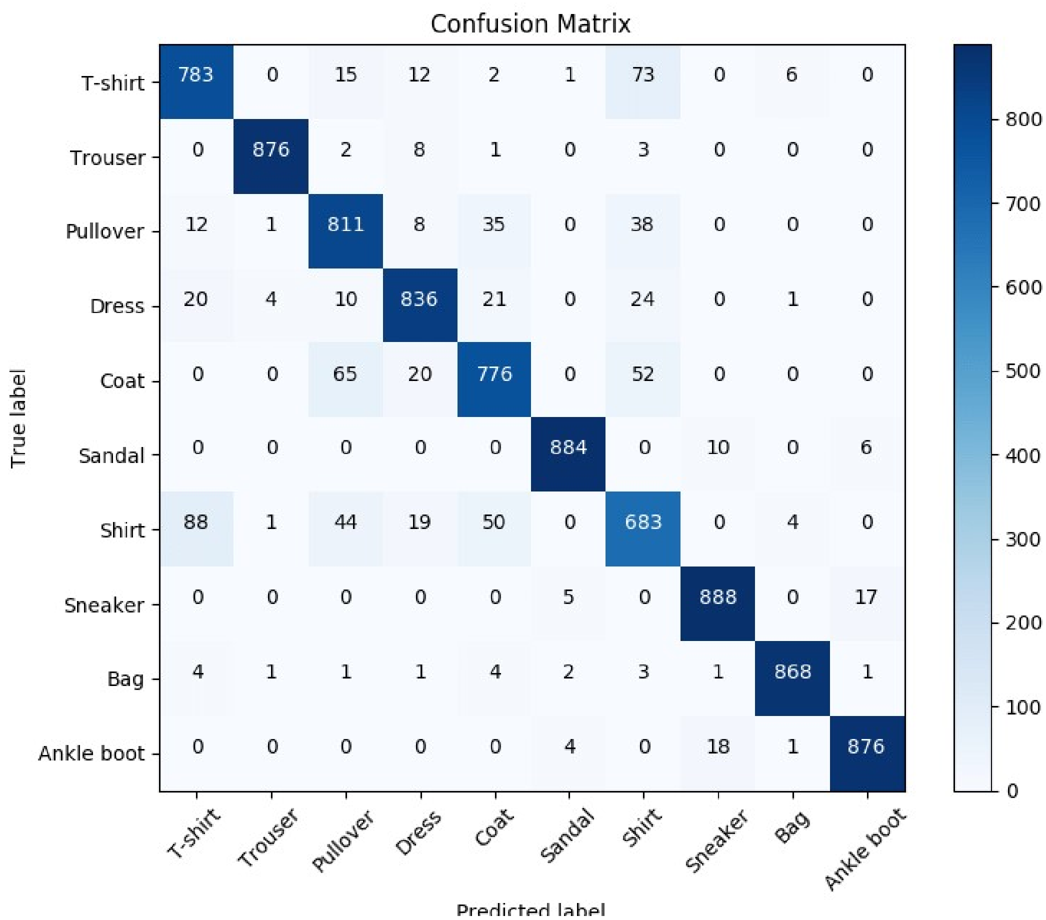
***3. Activation Function***

Activation function is just a thing that should be added to the output at the end of any neural network. This is used to obtain the output of the neural network like yes or no. Depending upon the function it maps the resulting values in between -1 to 1 or 0 to 1 etc. ReLU is really popular in the last few years and it is used in this model**.**

## Conculsion:

### *Fashion-MNIST Dataset*

In this work, we use F-MNIST dataset, which contains of 60,000 images of training and 10,000 images of test. Each gray scale image has a dimension of 28-by-28 pixels and grouped into ten categories from T-shirt/top to Ankle boots as displayed in two sample images in F-MNIST dataset, pictures of shoe.




We present some state-of-the-arts results to form a benchmark for F-MNIST. All neural network architecture results with several ConvNets models configuring hyperparameters and applying regularization are shown in below. SVC (Support Vector Classifier) is applied. Shows that these literature results with our best ConvNet performance.

With optimization used with four layer ConvNets we were capable of attain an accuracy of 91.3%. We can clearly see how by tuning various hyperparameters like optimizers, batch size, number of epochs and regularization methods such as image augmentation and dropout increase the overall performance and significantly decrease the training time. Fashion-MNIST can be a best drop-in substitution for MNIST although it is more difficult than MNIST dataset. We can implement or serve these models and regularization techniques for various types of image classification tasks and this dataset should be very much challenging when doing machine learning tasks.