**Conquering Fashion MNIST with CNNs using Computer Vision**

**ABOUT FASHION MNIST:**

The Fashion MNIST dataset is a popular benchmark dataset for image classification tasks in machine learning. It serves as an alternative to the original MNIST handwritten digits dataset and is often used to evaluate and compare the performance of different algorithms and models.

The Fashion MNIST dataset consists of a collection of 60,000 grayscale images of fashion items, such as clothing and accessories, divided into 10 classes. Each image is a 28x28 pixel square, resulting in a total of 784 pixels per image. The dataset is split into a training set of 50,000 images and a test set of 10,000 images.

The goal of the Fashion MNIST dataset is to develop machine learning models that can accurately classify the images into their respective classes. It is widely used for educational purposes, benchmarking models, and exploring various computer vision techniques. It can be classified using Tensorflow and Pytorch Package. In this case we will be using Tensorflow to create a CNN model.

**PROGRAM** :

#Importing libraries

import tensorflow as tf

from tensorflow.keras.datasets import fashion\_mnist

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D

from tensorflow.keras.layers import MaxPool2D

from tensorflow.keras.layers import Flatten

from tensorflow.keras.layers import Dropout

from tensorflow.keras.layers import Dense

import os

#Creating threads to decrease epoch time

os.environ['MKL\_THREADING\_LAYER'] = 'GNU'

print(tf.config.threading.get\_intra\_op\_parallelism\_threads())

#Reading and splitting dataset

(X\_train,y\_train) , (X\_test,y\_test)=fashion\_mnist.load\_data()

#Reshaping data

X\_train = X\_train.reshape((X\_train.shape[0], X\_train.shape[1], X\_train.shape[2], 1))

X\_test = X\_test.reshape((X\_test.shape[0],X\_test.shape[1],X\_test.shape[2],1))

print(X\_train.shape)

print(X\_test.shape)

#Normalizing data

X\_train=X\_train/255

X\_test=X\_test/255

#Creating Model

model=Sequential()

#Adding layers into model

model.add(Conv2D(32,(3,3),activation='relu',input\_shape=(28,28,1)))

model.add(MaxPool2D(2,2))

model.add(Flatten())

model.add(Dense(100,activation='relu'))

model.add(Dense(10,activation='softmax'))

#Compiling model

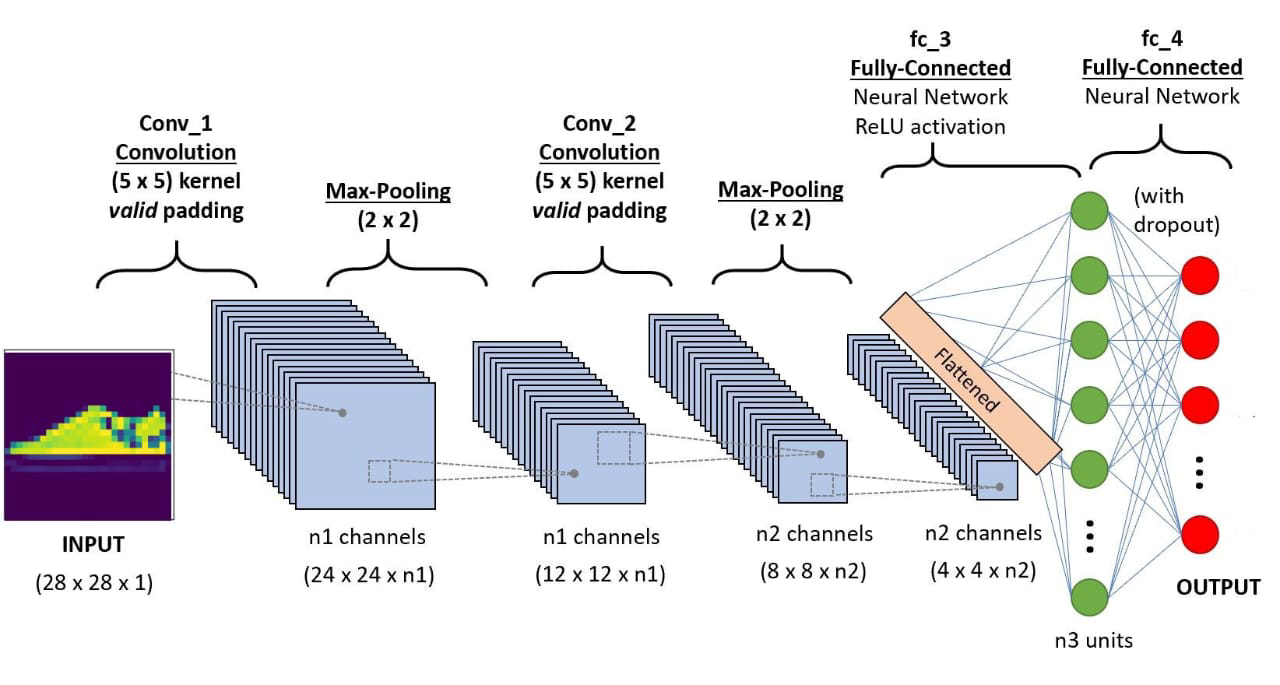
model.compile(loss='sparse\_categorical\_crossentropy',optimizer='adam',metrics=['accuracy'])

#Training Model

model.fit(X\_train,y\_train,epochs=10)

#Printing the loss and accuracy

Model.evaluate(X\_test, Y\_train)



**ABOUT THE PROGRAM:**

In this program we have used Tensorflow to create a CNN Model to classify the images in Fashion Mnist dataset. We have to download mkl and tensorflow package to run this above program. The Dataset contains 28X28 grayscale images. First we must split the data into testing and training dataset. The X part contains the images and the Y part contains the type of cloth. We first convert the X parts into binary form black are represented using 0 and white are represented using 1. Next we create a CNN model.

This CNN model has a convolution layer, pooling layer, and a fully connected layer. In the convolution layer we first multiply the matrix X with a filter to recognize the patterns in the image like lines, curves etc. The Multiplied matrix is called the activation map. This activation map is then used to find the confidence score of each image. The confidence score says the probability of a certain image to belong to a class. Then Max Pooling is done to further reduce the given matrix. After which the weights are adjusted to get the desired output.

We were able to achieve an accuracy of 90% without any optimization and each epoch took around 40 seconds to execute. We were able to increase the accuracy to 91.6% and reduce the time taken to execute each epoch to 20 seconds. We will try to further improve this model in the coming week our aim is to increase the accuracy to 93%.

**PROGRAM OUTPUT :**

[**https://drive.google.com/file/d/1NhVxdrShIz9suhV-6E7keu4axxwzXVy8/view?usp=drivesdk**](https://drive.google.com/file/d/1NhVxdrShIz9suhV-6E7keu4axxwzXVy8/view?usp=drivesdk)