**PROJECT REPORT**

**FASHION MNIST CLASSIFICATION**

**1. Project Objective:**

The goal of this project is to build and train a Convolutional Neural Network (CNN) to accurately classify images from the Fashion MNIST dataset into 10 different fashion categories.

**2. Dataset Used:**

**Dataset**: Fashion MNIST

**Description**: This dataset is a collection of 70,000 grayscale images (28x28 pixels) of articles of clothing and accessories. It is divided into:

**Training set:** 60,000 images

**Testing set:** 10,000 images

**Classes (10 total):** T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot.

**3. Data Preprocessing:**

**Normalization:** The pixel values of the images were scaled from the original range of 0-255 to a range of 0-1 by dividing by 255. This helps in faster and more efficient training of the model.

**Reshaping:** The input images were reshaped from (60000, 28, 28) to (60000, 28, 28, 1) to make them compatible with the CNN model which expects a channel dimension.

**One-Hot Encoding:** The target labels (y\_train and y\_test) were converted into a one-hot encoded format. For example, a label of '3' was converted to [0, 0, 0, 1, 0, 0, 0, 0, 0, 0].

**4. Model Architecture:**

A Sequential CNN model was constructed with the following layers:

**Conv2D Layer:** 32 filters, kernel size (3,3), ReLU activation.

**MaxPooling2D Layer:** (2,2) to downsample the feature maps.

**BatchNormalization:** To normalize the outputs of the previous layer.

**Conv2D Layer:** 64 filters, kernel size (3,3), ReLU activation.

**MaxPooling2D Layer:** (2,2).

BatchNormalization.

**Flatten Layer:** To convert the 2D feature maps into a 1D vector.

**Dense Layers:** A series of fully connected layers with 250, 128, 64, and 32 neurons, all using ReLU activation.

**Dropout Layers:** Dropout with a rate of 0.35 was applied after each dense layer to prevent overfitting.

**Output Layer:** A Dense layer with 10 neurons and a 'softmax' activation function to output a probability distribution over the 10 classes.

**5. Model Training and Evaluation:**

**EarlyStopping:** The training was set to stop if the validation loss did not improve for 5 consecutive epochs.

**ModelCheckpoint:** The best version of the model (based on validation loss) was saved during training.

**Compilation:** The model was compiled using the Adam optimizer and categorical\_crossentropy as the loss function, with accuracy as the evaluation metric.

**Training:** The model was trained for 20 epochs with a batch size of 64 and a validation split of 35%.

**6. Performance**

The trained model achieved the following performance:

**Test Accuracy:** 90%

**Test Loss:** 0.33

**Train Accuracy:** 92%

**Train Loss:** 0.23

The classification report showed good precision and recall across most classes.

**7. Custom Input Prediction**

A function was created to process custom images (e.g., a jacket, a sneaker) to match the Fashion MNIST format (28x28 grayscale). The trained model was then able to correctly predict the class for these custom images.

For an image of a jacket, the model predicted the label Coat.

For an image of a sneaker, the model predicted the label Sneaker.zer='he\_uniform', activation='softmax'))