# **CNN Capstone Project: Fashion-MNIST Classification**

## **Project Overview**

Build a Convolutional Neural Network (CNN) to classify clothing items from the Fashion-MNIST dataset. This project provides an excellent introduction to computer vision and deep learning with a more practical and relatable dataset than traditional handwritten digits.

## **Dataset Description**

**Fashion-MNIST** is a dataset of clothing article images, serving as a modern replacement for the classic MNIST digit dataset:

* **Classes (10 categories)**:  
  + 0: T-shirt/top
  + 1: Trouser
  + 2: Pullover
  + 3: Dress
  + 4: Coat
  + 5: Sandal
  + 6: Shirt
  + 7: Sneaker
  + 8: Bag
  + 9: Ankle boot
* **Image specifications**: 28×28 pixels, grayscale (1 channel)
* **Training set**: 60,000 images (6,000 per class)
* **Test set**: 10,000 images (1,000 per class)
* **Availability**: Built into Keras datasets (tensorflow.keras.datasets.fashion\_mnist)

## 

## **Project Steps**

### **Step 1: Data Loading and Exploration**

* Load Fashion-MNIST dataset using Keras
* Examine dataset structure and class distribution
* Visualize sample images from each clothing category
* Analyze pixel value distributions and image characteristics
* Create class name mappings for better interpretability

**Deliverable**: Exploratory data analysis notebook with visualizations

### **Step 2: Data Preprocessing**

* Normalize pixel values to [0,1] range
* Reshape data for CNN input (add channel dimension)
* Convert labels to categorical (one-hot encoding)
* Split training data into train/validation sets (80/20 split)
* Implement data augmentation (rotation, width/height shift, zoom)

**Deliverable**: Preprocessed datasets ready for training

### **Step 3: Baseline Model**

* Create a simple feedforward neural network as baseline
* Train and evaluate baseline performance
* Document baseline accuracy and limitations
* This establishes the minimum performance threshold

**Deliverable**: Baseline model results and analysis

### **Step 4: CNN Architecture Design**

Design a CNN with progressive complexity:

**Simple CNN Architecture**:

* Conv2D layer (32 filters, 3×3 kernel) + ReLU + MaxPooling
* Conv2D layer (64 filters, 3×3 kernel) + ReLU + MaxPooling
* Flatten + Dense(128) + Dropout(0.5)
* Dense(10) + Softmax

**Advanced CNN Architecture**:

* Multiple convolutional blocks with batch normalization
* Increasing filter sizes (32 → 64 → 128)
* Global Average Pooling instead of flattening
* Regularization techniques

**Deliverable**: Working CNN architectures with clear documentation

### **Step 5: Model Training and Optimization**

* Implement training loop with appropriate callbacks
* Use techniques like:
  + Early stopping to prevent overfitting
  + Learning rate scheduling
  + Model checkpointing
* Experiment with different optimizers (Adam, SGD)
* Fine-tune hyperparameters (learning rate, batch size, epochs)

**Deliverable**: Trained models with optimization history

### **Step 6: Model Evaluation and Analysis**

* Evaluate model on test set
* Create confusion matrix to identify misclassified categories
* Generate classification report with precision, recall, F1-scores
* Visualize training/validation curves
* Analyze which clothing items are most/least accurately classified
* Compare CNN performance with baseline model

**Deliverable**: Comprehensive evaluation report