FASHION MNIST DATA USING CNN

1. Problem Statement

The goal of this project is to classify images of clothing items from the **Fashion MNIST dataset**. Unlike the original MNIST dataset of handwritten digits, Fashion MNIST provides a more challenging benchmark for image classification.
The problem is a **supervised multi-class classification task** with 10 output categories (T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot).

2. Dataset Overview

- **Dataset**: Fashion MNIST (Kaggle).
- **Size**: 70,000 grayscale images (28×28 pixels).
    - Training: 60,000
    - Testing: 10,000
- **Classes**: 10 categories of clothing items.
- **Format**: Each image is a 28×28 array of pixel values (0–255). Labels are integers from 0–9, each representing a category.

3. List of Libraries/Packages Used

- **Python core**: numpy, pandas
- **Visualization**: matplotlib, seaborn
- **Machine Learning / Deep Learning**: tensorflow, keras
- **Utilities**: os, warnings

4. Preprocessing Steps

- **Preprocessing**:
    - Normalization: pixel values scaled from [0, 255] → [0, 1].
    - Splitting: dataset divided into training and testing sets.
    - Encoding: class labels transformed into categorical one-hot vectors.

5. Methodology

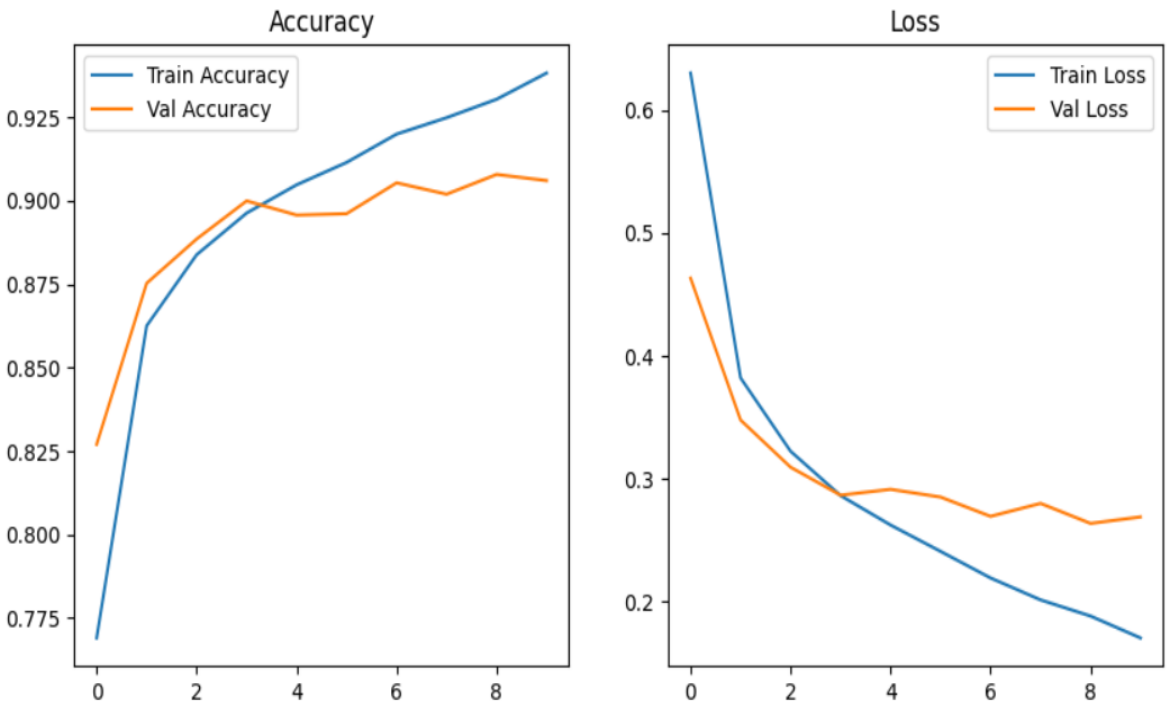
- **Data Cleaning**:
  - Ensured dataset integrity (no missing values).
  - Scaled features for stable neural network training.
- **Feature Engineering:**
  - Images reshaped and normalized.
- **Model Building**:
  - Built a **Convolutional Neural Network (CNN)** using Keras.
  - Typical structure: convolutional layers (for feature extraction), pooling layers (for dimensionality reduction), fully connected layers (for classification).
  - Used **Softmax** activation in the output layer for multi-class classification.
  - We use **argmax** to map model probability outputs → discrete class labels, so we can evaluate accuracy and interpret predictions.

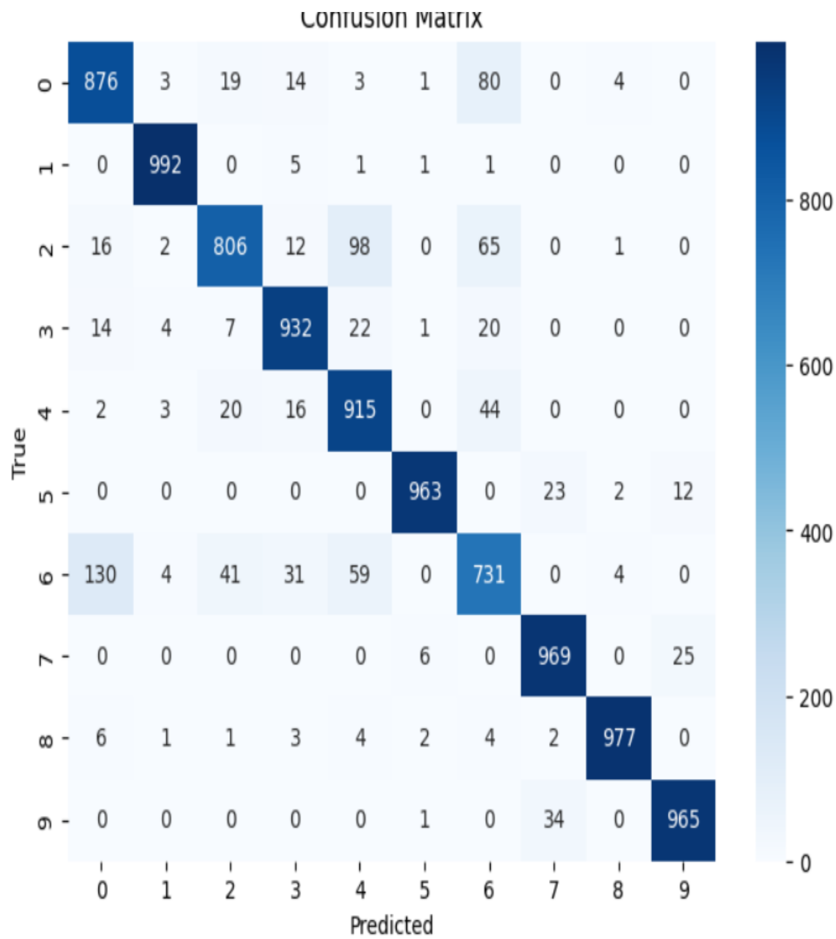
6. Evaluation Metrics

- **Accuracy**: Primary evaluation metric [accuracy, precision, recall, F1 score for classification report]
- **Loss (Cross-Entropy)**: Monitored during training.
- **Confusion Matrix**: Used to evaluate per-class performance.

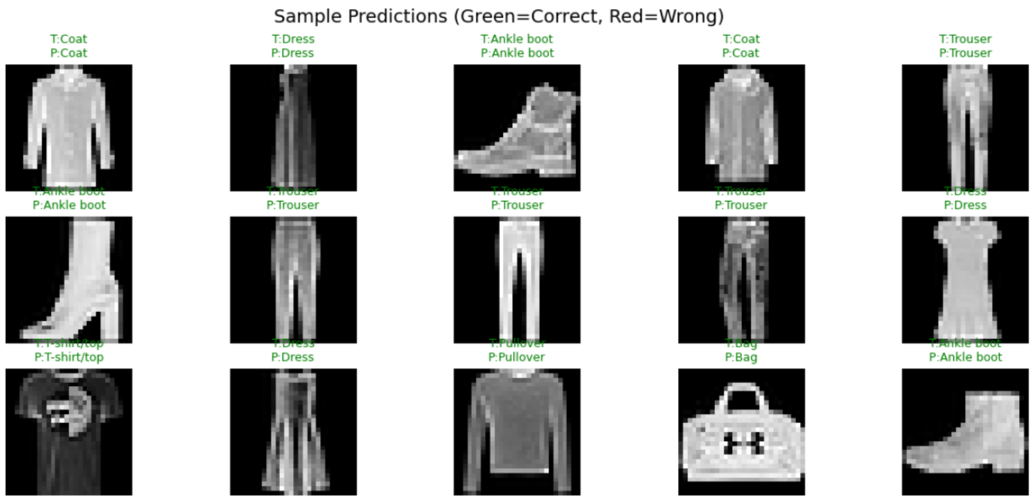
7. Results & Analysis

- **Training vs Validation**: Plotted loss and accuracy curves across epochs.
- **Model Accuracy**: Typically CNN models on Fashion MNIST achieves **91% test accuracy**.
- **Visualization**:
  - Plots of training history (accuracy/loss).
  - Sample test images with predicted labels.
  - Misclassified examples highlighted to show where the model struggles (e.g., Shirt vs T-shirt)


Accuracy
Train Accuracy
Val Accuracy
0.925
0.900
0.875
0.850
0.825
0.800
0.775
0 2 4 6 8
Loss
Train Loss
Val Loss
0.6
0.5
0.4
0.3
0.2
0 2 4 6 8


Confusion Matrix
True
Predicted
0: 876 3 19 14 3 1 80 0 4 0
1: 0 992 0 5 1 1 1 0 0 0
2: 16 2 806 12 98 0 65 0 1 0
3: 14 4 7 932 22 1 20 0 0 0
4: 2 3 20 16 915 0 44 0 0 0
5: 0 0 0 0 0 963 0 23 2 12
6: 130 4 41 31 59 0 731 0 4 0
7: 0 0 0 0 0 6 0 969 0 25
8: 6 1 1 3 4 2 4 2 977 0
9: 0 0 0 0 0 1 0 34 0 965
0 1 2 3 4 5 6 7 8 9
800
600
400
200
0

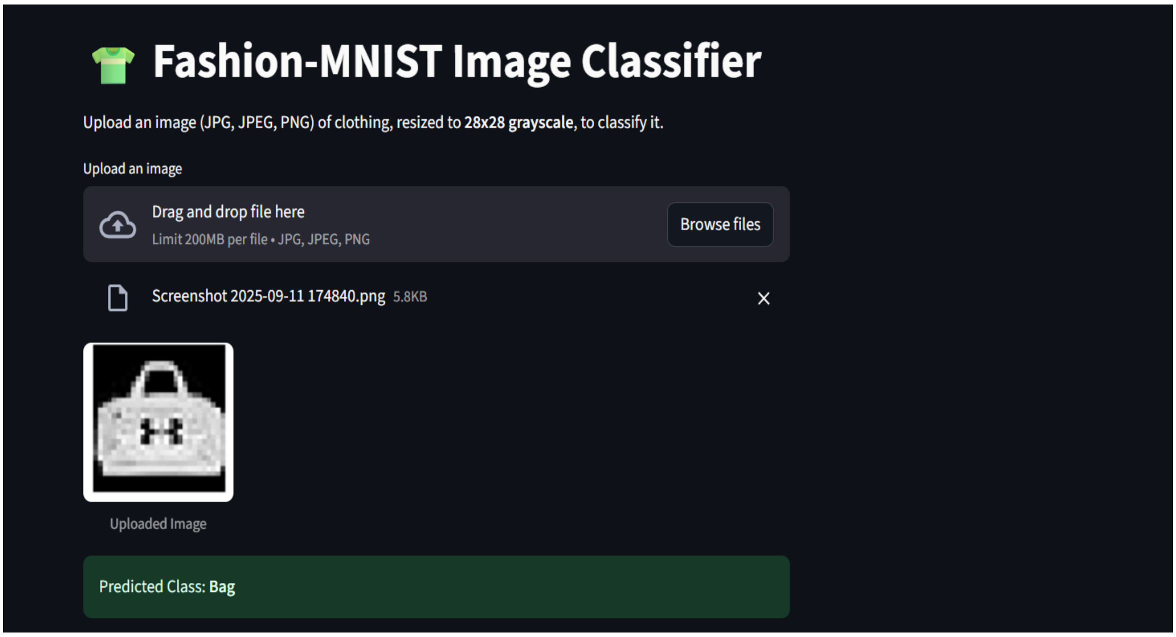
```
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.88      0.86      1000
           1       0.98      0.99      0.99      1000
           2       0.90      0.81      0.85      1000
           3       0.92      0.93      0.93      1000
           4       0.83      0.92      0.87      1000
           5       0.99      0.96      0.98      1000
           6       0.77      0.73      0.75      1000
           7       0.94      0.97      0.96      1000
           8       0.99      0.98      0.98      1000
           9       0.96      0.96      0.96      1000

    accuracy                           0.91     10000
   macro avg       0.91      0.91      0.91     10000
weighted avg       0.91      0.91      0.91     10000
```



The Fashion MNIST dataset was integrated and deployed via GitHub and Streamlit Community to build an interactive web application for real-time image classification

1. Github Link: https://github.com/vijithtechverse/Fashion_Mnist-

2. Streamlit App: https://8swunfqzjfc6v4jgjwyrpy.streamlit.app/



8. Conclusion

- CNN successfully classified Fashion MNIST images with high accuracy.
- Demonstrated the effectiveness of deep learning for image classification tasks.
- Key limitations: some confusion between visually similar classes.
- Future improvements:
    - Data augmentation for better generalization.
    - Experimenting with deeper CNNs, transfer learning, or attention mechanisms.