Project Report

End-to-End Image Classification System for Fashion-MNIST using Custom CNN and Streamlit

Course Name: Neural Networks

Team Members:

[Ahmed Ashraf]

[Ahmed Hassan]

[Abdelrahman Medhat]

[Mahmoud Adel]

[Wageeh Hussain]

Date: December 2025

1. Abstract

This project presents a complete Deep Learning pipeline for classifying images from the Fashion-MNIST dataset. A custom Convolutional Neural Network (CNN) was developed to accurately distinguish between 10 categories of clothing. To bridge theory and practice, the trained model was deployed as an interactive Streamlit web application that allows users to upload images and receive real-time predictions with confidence scores.

2. Introduction

The Fashion-MNIST dataset is a modern benchmark consisting of 70,000 grayscale images (28×28 pixels) across 10 clothing classes. This project aims to build a high-performance CNN model and demonstrate its effectiveness through a user-friendly interface.

3. Problem Statement

Traditional machine learning techniques struggle to capture spatial patterns in image data. This project addresses this limitation using deep learning to automatically extract features from raw pixels, classify clothing items accurately, and present predictions through a graphical user interface.

4. Methodology

The system was developed using structured data preprocessing, augmentation, custom CNN design, and optimized training strategies.

4.1 Data Preprocessing

Pixel normalization to the [0,1] range, reshaping to (28,28,1), one-hot encoding of labels, and splitting the dataset into training (90%) and validation (10%) sets.

4.2 Data Augmentation

Real-time augmentation techniques such as random rotations, width and height shifts, horizontal flipping, shearing, and zooming were applied.

4.3 Model Architecture

The CNN consists of three convolutional blocks with 64, 128, and 256 filters. Each block includes Conv2D layers, Batch Normalization, ReLU activation, MaxPooling, and Dropout. Global Average Pooling and a Dense Softmax layer complete the architecture.

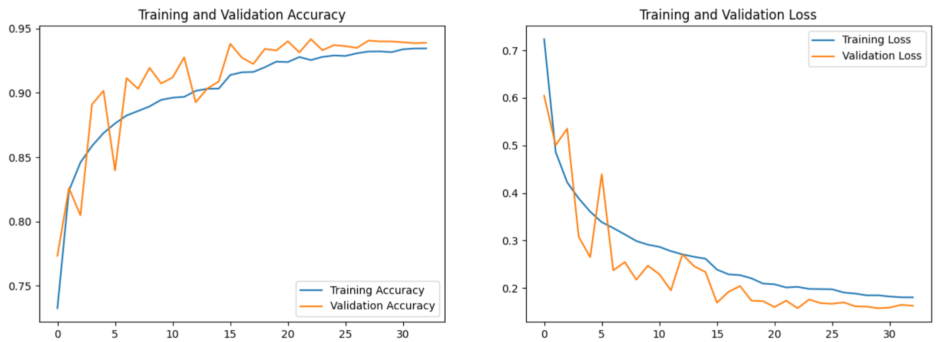
4.4 Training Configuration

The model was trained using the Adam optimizer with a learning rate of 0.001 and Categorical Crossentropy loss. EarlyStopping and ModelCheckpoint callbacks were used to preserve optimal model weights.

5. Experimental Results

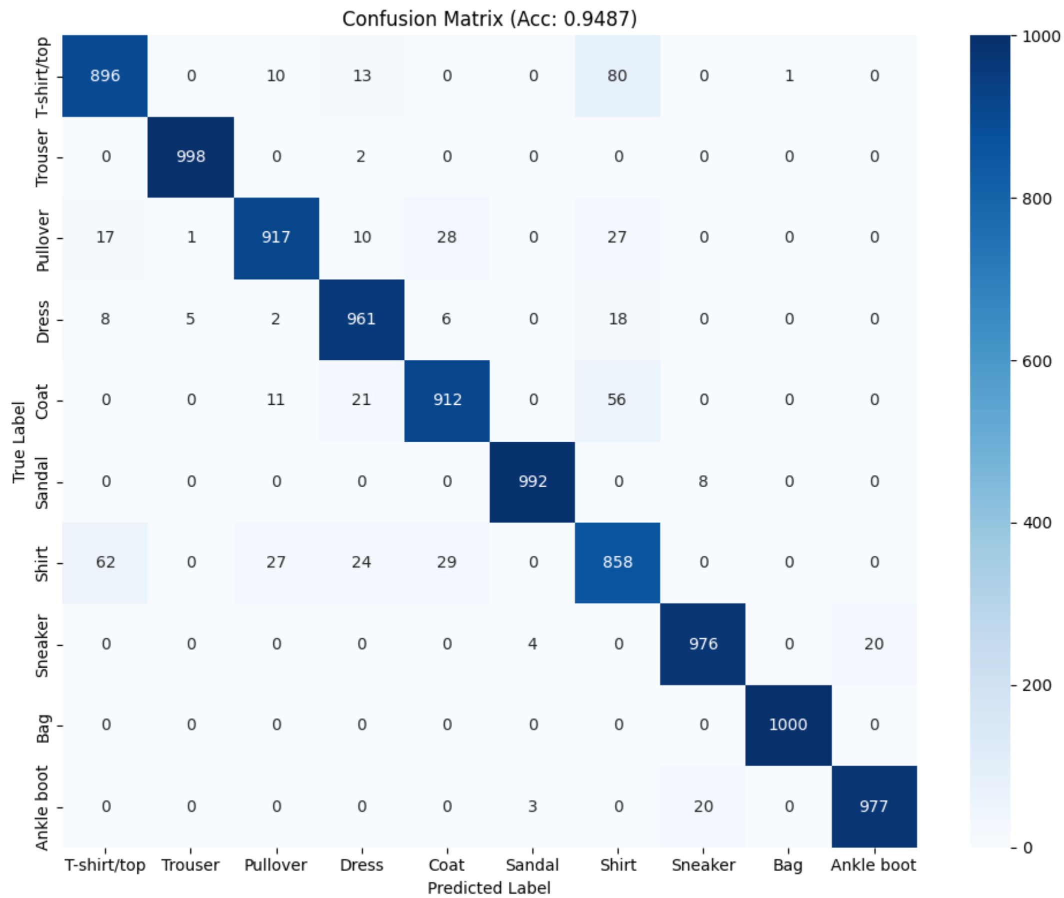
5.1 Training Curves

This figure shows training and validation accuracy and loss over epochs.



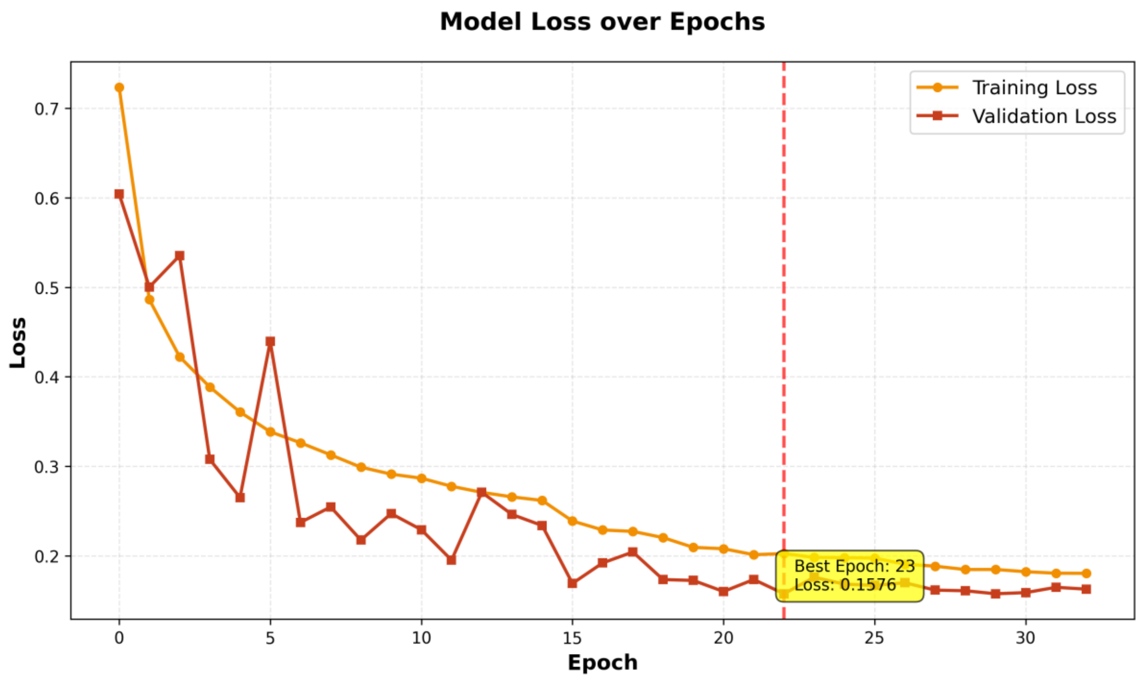
5.2 Confusion Matrix

The confusion matrix illustrates classification performance across all classes.



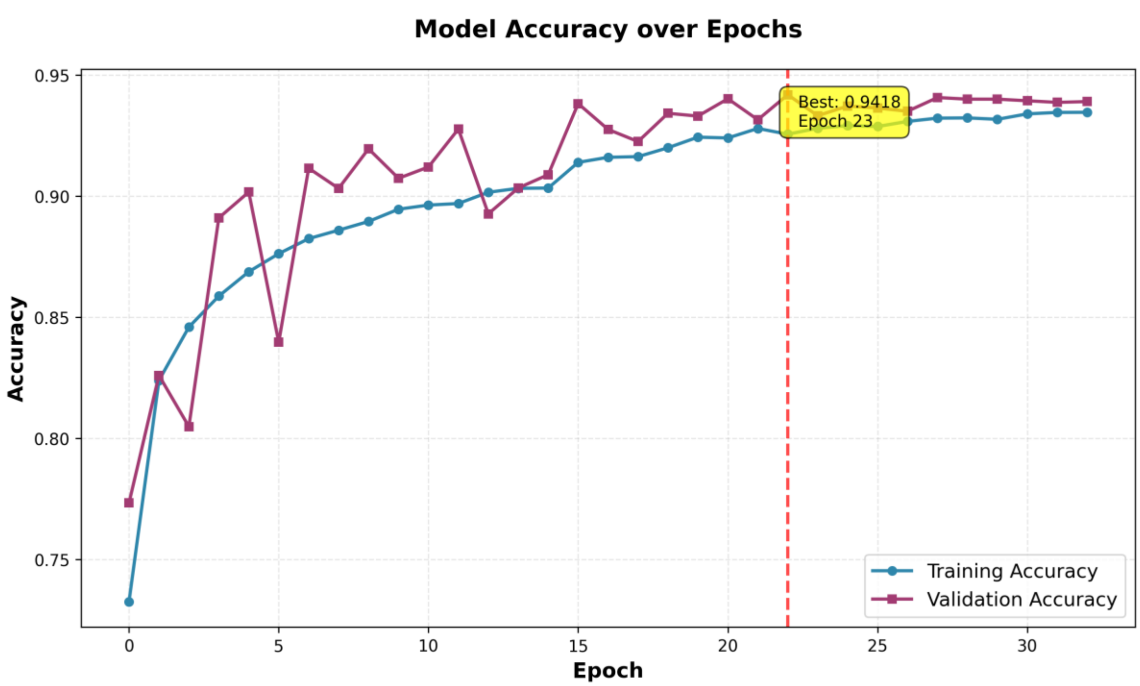
5.3 Loss Curve

This plot highlights convergence behavior during training.



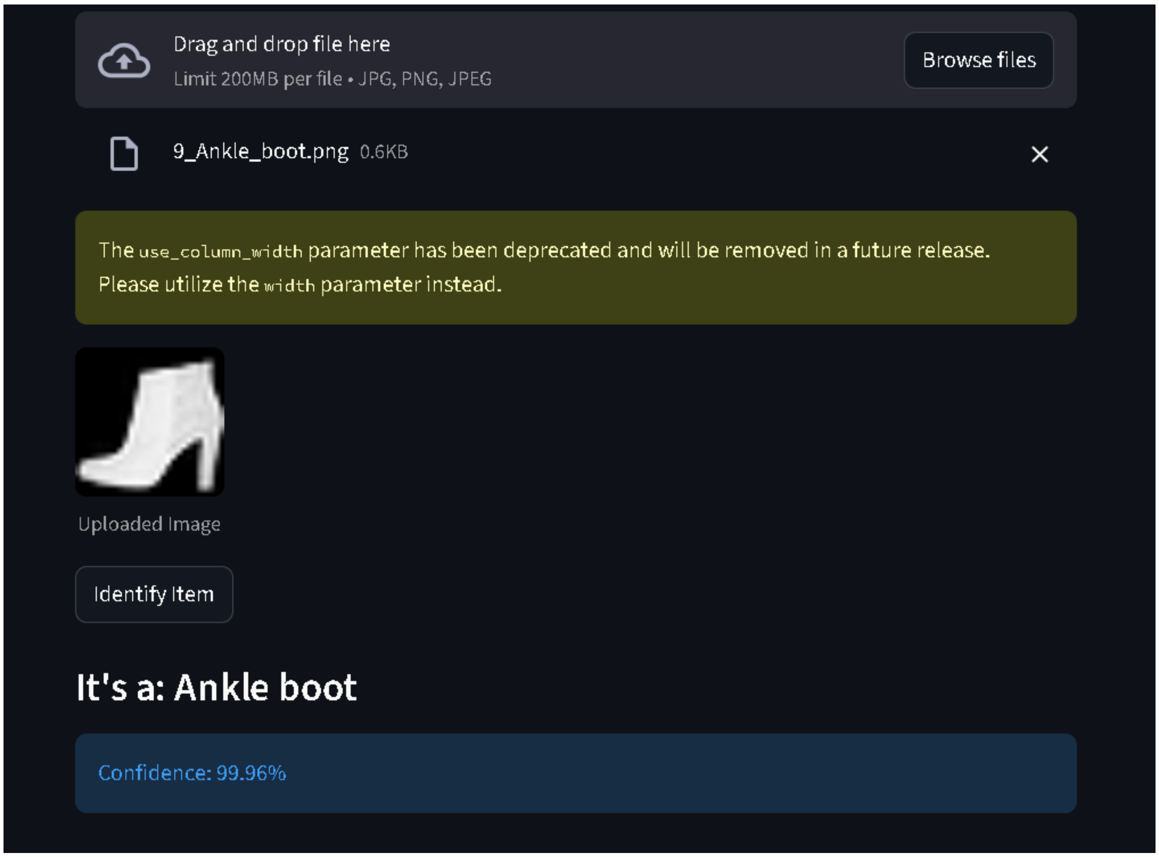
5.4 Accuracy Curve

This plot shows accuracy trends across epochs.



5.5 Streamlit Interface Screenshot

Screenshot of the deployed Streamlit web application.



6. Conclusion

The project successfully delivered a robust CNN-based image classification system with strong performance. Data augmentation and batch normalization improved generalization, while Streamlit deployment demonstrated practical usability beyond experimental evaluation.

7. Technologies Used

Python 3.10, TensorFlow (Keras), NumPy, Pandas, OpenCV, Matplotlib, Seaborn, Streamlit, Git, GitHub.