Deep Learning Classifier for Fashion-MNIST

From Raw Data to Deployed Streamlit Web App

FacultyofScience,CairoUniversity - CSDepartment

The Problem & Motivation

Traditional algorithmsstrugglewithimagecomplexity,especiallywhen visual differences are subtle.

The Challenge: Fashion-MNIST

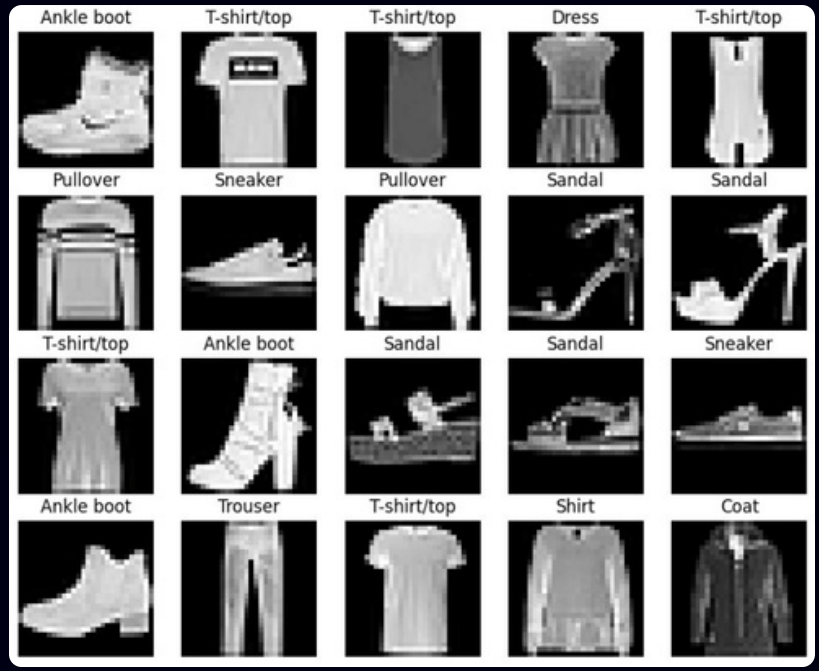
A dataset of 10 clothing classes (e.g., Shirt, Coat, Pullover) where items are visually very similar, making differentiation difficult for machines.

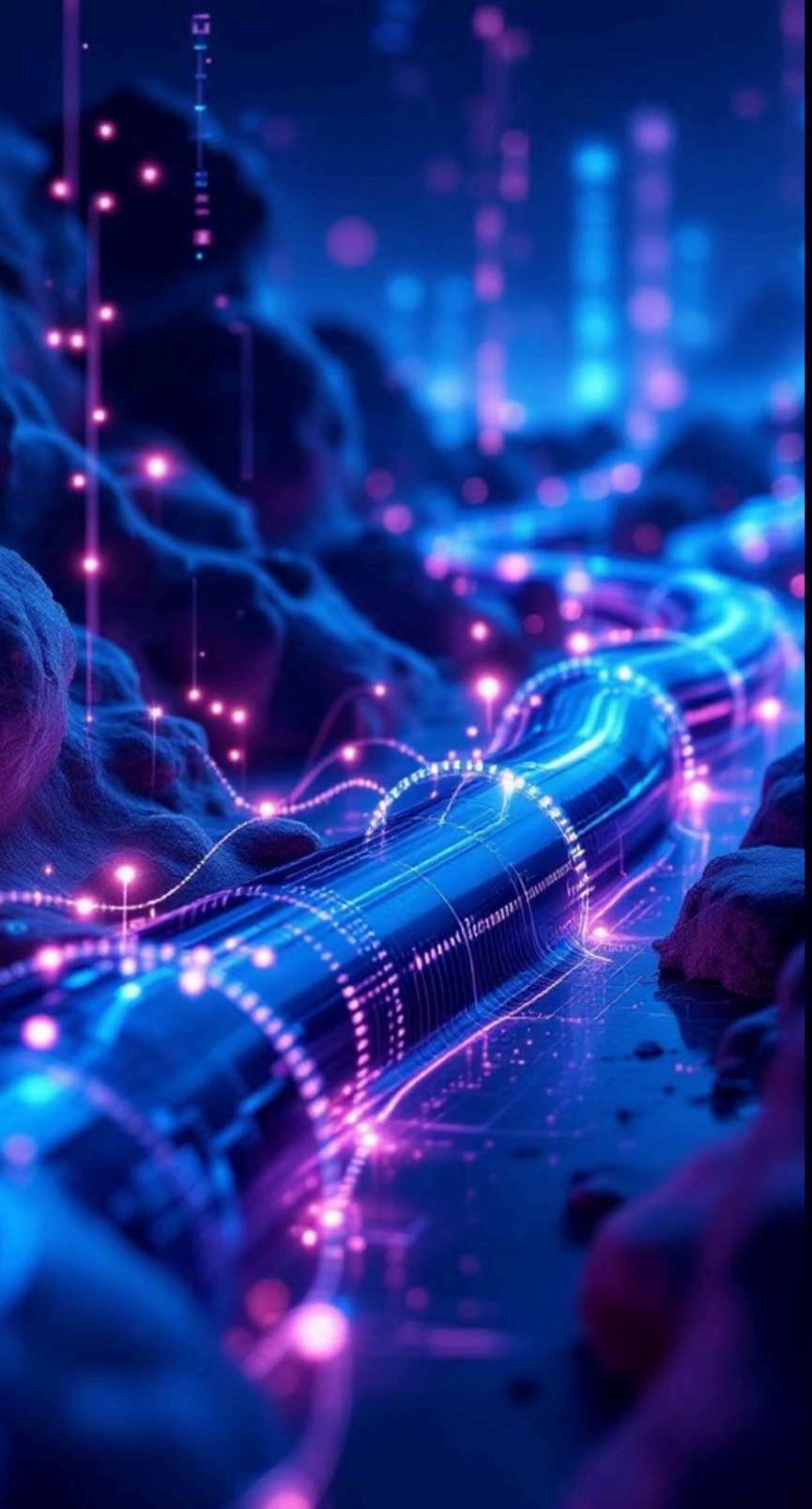
Our Goal

Develop a robust AI model for high-accuracy classification and deploy it for practical, real-world application.

TARGET USERS & IMPACT: E-COMMERCE

Platforms can auto-tag user-uploaded products, improving search and recommendation engines.





Data Pipeline & Preprocessing

01

Dataset Acquisition

70,000 grayscale images, each 28x28 pixels, from the Fashion-MNIST collection.

02

Normalization

Pixel values scaled from 0-255 to 0-1, enhancing model stability and training speed.

03

Reshaping & Encoding

Inputs transformed to (28, 28, 1) for CNN compatibility. Labels converted via One-Hot Encoding.

04

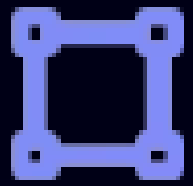
Data Augmentation

Utilized ImageDataGenerator for random rotations, zooms, and shifts to combat overfitting and improve generalization.

Why Is This Dataset Hard?

Subtle Differences

Unlike distinct digits (0-9), clothing items like a "Shirt" and "T-shirt" share very similar structural features.



Low Resolution

At only 28×28 pixels, fine details (textures, buttons, logos) are lost, reducing the feature set available.

No Color Info

Being grayscale means the model cannot rely on color patterns, forcing it to learn purely from shape and edge data.

Model Architecture: The Custom CNN Core

Input Layer

Receives28x28x1 grayscale images.

Convolutional Blocks (x3)

- **FeatureExtraction:** Conv2D layers with increasing filters (64 -> 128 -> 256) to capture hierarchical features.
- **Stability & Activation:** BatchNormalization and ReLU.
- **Downsampling:** MaxPooling2D for dimensionality reduction.
- **Regularization:** Dropout layers to prevent overfitting.

The Head: GlobalAveragePooling2D

Efficiently flattensfeaturemaps,reducingparameters compared to traditional Flatten layers.

Output Layer

Denselayer withSoftmax activation for 10-class probability distribution.

Training Strategy: Optimizing Performance

Optimizer: Adam

Anadaptivelearning rate optimization algorithm known for its efficiency and effectiveness in deep learning.

Loss Function: Categorical Crossentropy

Idealformulti-class classification, measuring the performance of a classification model whose output is a probability value between 0 and 1.

Callbacks for Smart Training

- **EarlyStopping:**Haltstraining when validation performance stops improving.
- **ReduceLROnPlateau:** Dynamically adjusts learning rate when a metric has stopped improving.

Evaluation & Results

Key Metrics

95%

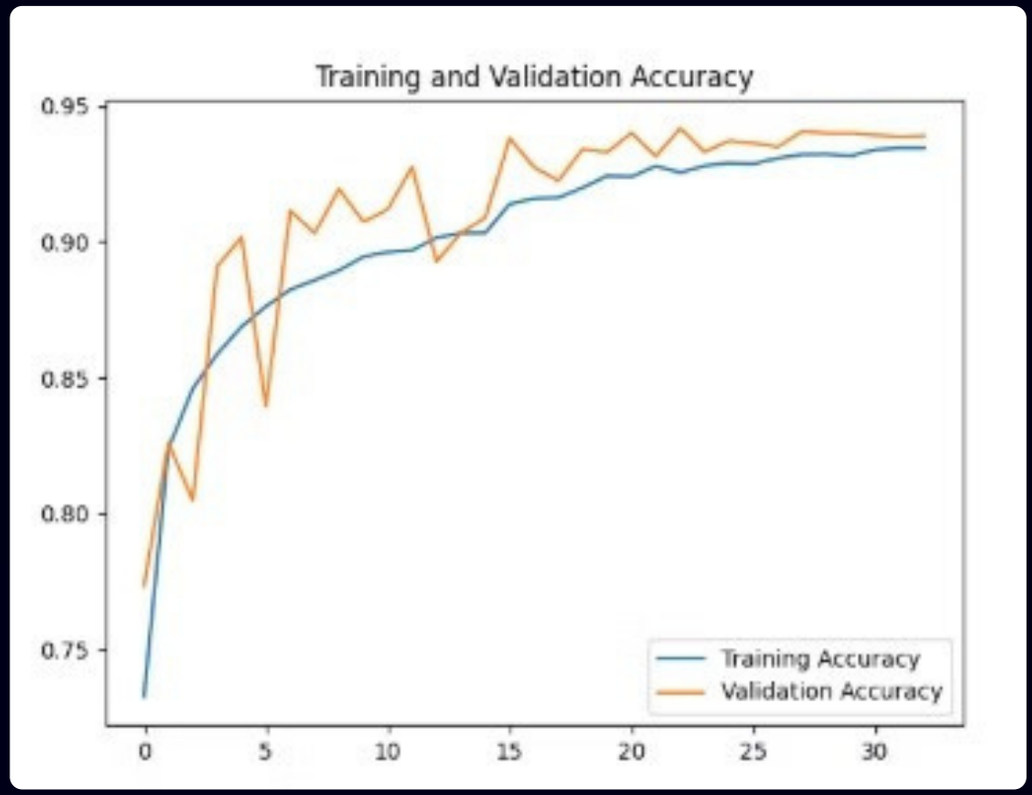
Test Accuracy

Achieved high accuracy on the unseen test dataset, validating model generalization.

Misclassification Analysis

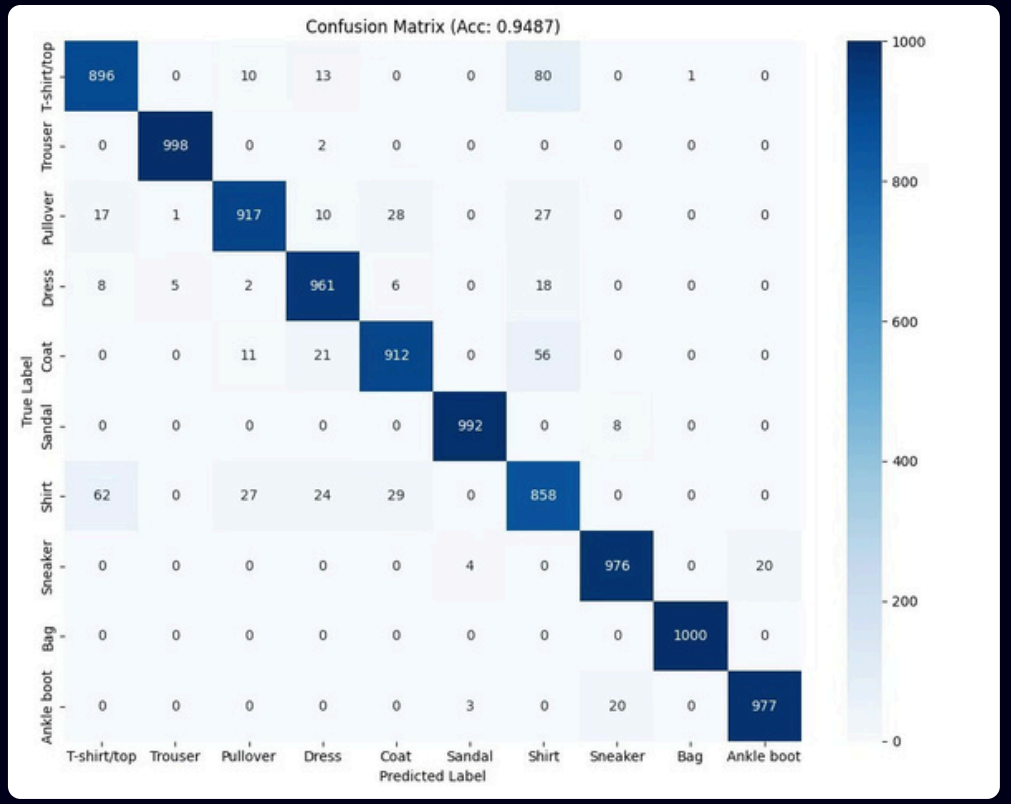
A **Confusion Matrix** revealed specific class confusions, notably between 'Shirts' and 'Coats', informing future model refinements.

Visual inspection of **Loss and Accuracy curves** confirmed stable convergence and absence of significant overfitting throughout training.



Confusion Matrix

A**Confusion Matrix** revealed specific class confusions, notably between '**Shirts**' and '**T-Shirts**', informing future model refinements.



Use Pre-Train Model ResNet50V2

Toimproveaccuracy despite datalimitations, we leveraged Transfer Learning.

- Why ResNet50V2? It has learned rich feature representations from the massive ImageNet dataset.
- Adaptation: We replaced the top layers to classify our 10 fashion categories instead of ImageNet's 1000 objects.
- Benefit: The model "knows" how to detect edges, shapes, and textures, even if the input is small.

We achieve 94% train accuracy with this model

HARDWARE LIMITATIONS

GOOGLE COLAB CONSTRAINTS

Runningadeep modellike ResNet50V2requires significant memory.

The Bottleneck: Pre-trained models expect standard input sizes more then (35×35). Upscaling 70k images to this size caused **RAM crashes** in the Colab environment.

The Fix: We had to find a "sweet spot" for image resizing that balanced model performance with available hardware resources.

EXPERIMENTAL APPROACH

STEP 1

Baseline
35×35 px

Original Size

STEP 2

Upscaling
48×48 px

Testing
Limits

STEP 3

Optimization
64×64 px

Better Features

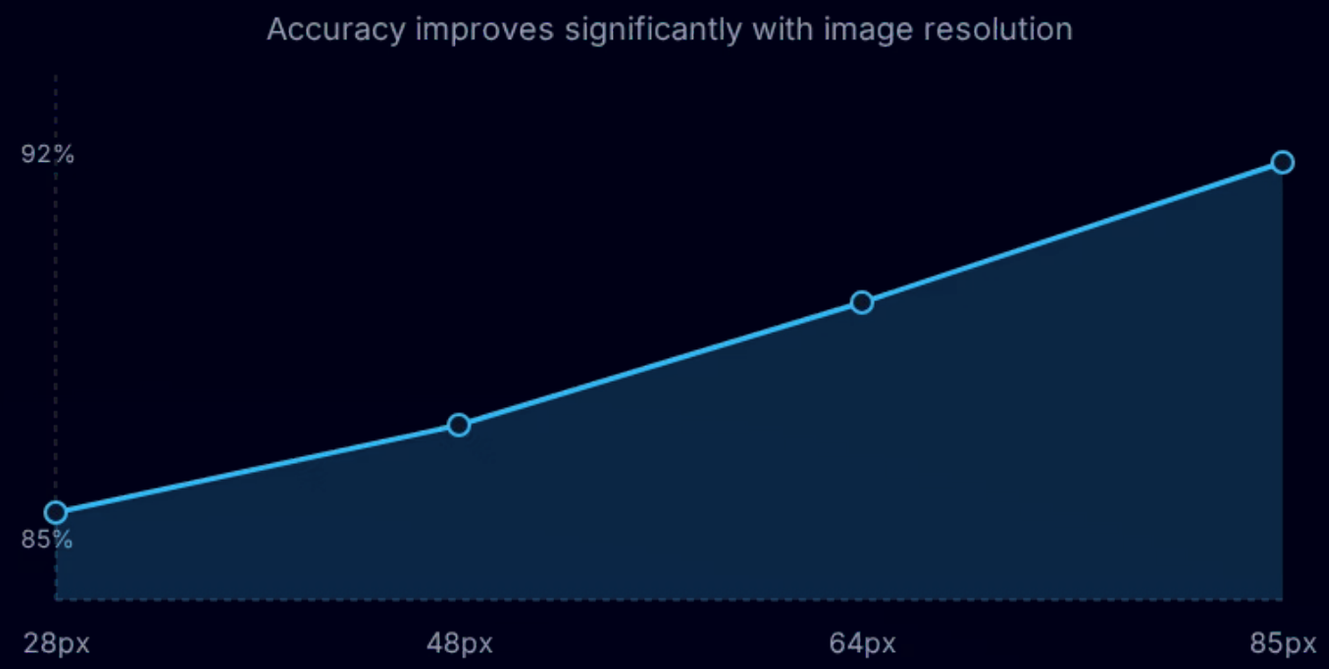
STEP 4

Final
85×85 px

Colab Max

We systematically increased the image input size to find the maximum limit before crashing.

RESULTS: SIZE VS. ACCURACY



Despite hardware limits, increasing the input size to 85x85 allowed ResNet50V2 to capture enough spatial detail to significantly boost classification accuracy.

Deployment: The Interactive Web App

Wetransformed ourrobust classification model intoan accessible and interactive web application using **Streamlit**.

Interactive UI

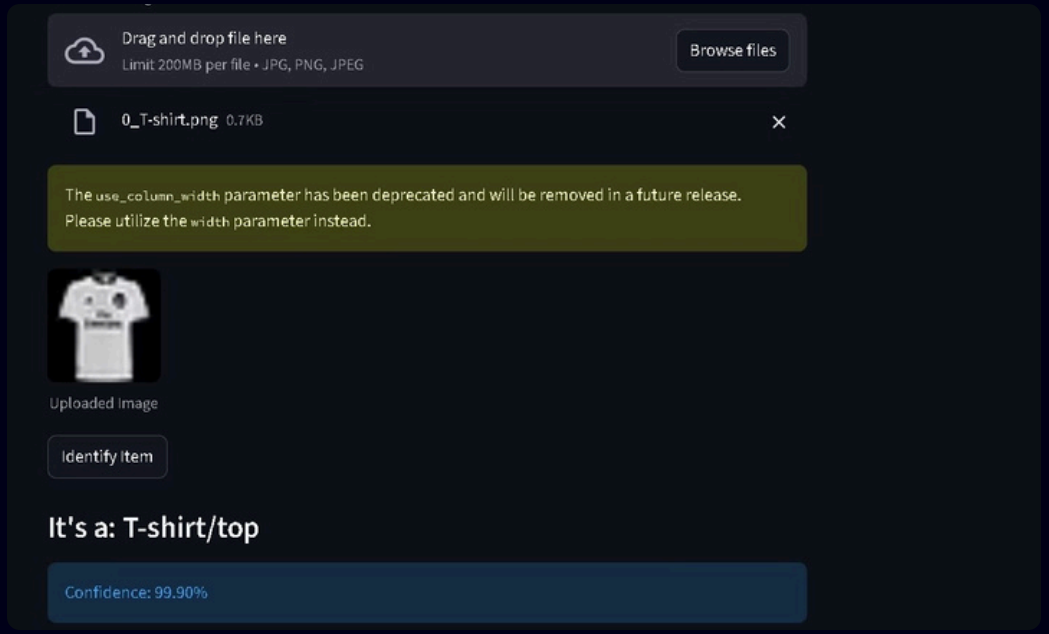
A custom-designed user interface allows seamless interaction and prediction.

Real-time Prediction

Users can upload any clothing image and receive an instant classification with confidence scores.

Probability Visualization

A sidebar displays prediction probabilities across all classes, offering transparency.



Technologies Used

Python 3.10

The core programminglanguage for model development and application logic.

TensorFlow (Keras)

Our primarydeep learning framework for building and training the CNN.

Pandas & NumPy

Essentiallibrariesforefficientdata manipulation and numerical operations.

OpenCV

Used for image processing tasks and handling image data.

Streamlit

The powerful framework behind our interactive web application deployment.

PyCharm

Our Integrated DevelopmentEnvironment for robust coding.

Git & GitHub

Versioncontroland collaborative development platform.

Google Colab

For trainandevaluatemodels

using GPU

Team Members

Wageeh Hussain

Abdelrahman Medhat

Ahmed Hassan

Mahmoud Adel

Ahmed Elkot