RIPHAH INTERNATIONAL UNIVERSITY

Short Project 02:

Smart Classifier for Fashion

Prepared BY:

Abrar Ali
BS-SE (5-2), Sap ID: 55843

Date:
Nov 14, 2025

Subject: Artificial Intelligence

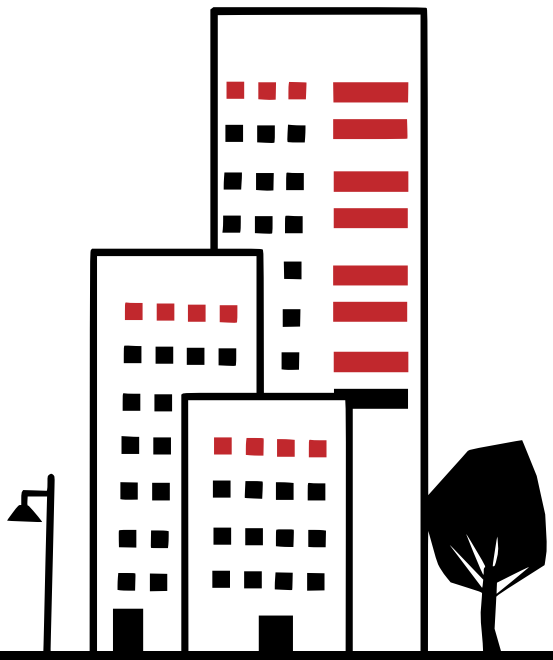


Table of Contents

1. Introduction: What Was Our Goal?

The goal of this project was to teach a computer how to tell the difference between 10 types of clothing (like shirts, bags, and shoes).

To do this, we used a special kind of "digital brain" called a **Convolutional Neural Network (CNN)**. This is a type of A.I. that is very good at "seeing" and understanding pictures.

Colab Link

2. The Data: What Did We Use to "Teach" the Computer?

We used the **Fashion-MNIST dataset**, a famous collection of **70,000 small, black-and-white pictures** of clothing.
We used **60,000 pictures to train** our model and **10,000 to test** it at the end.

Before we started, we had to prepare the pictures. We **"normalized"** them (scaling all the pixel values to be between 0 and 1) so the computer could learn from them more easily.

Here are some examples of what the pictures look like:

3. The Model: Our "Digital Brain's" Blueprint

We built our CNN model in **layers**, or "cells."
You can think of it as an assembly line, where each layer has a specific job.

1. **Convolutional Layers (The "Scanners")**

   These first layers scan the image to find simple patterns, like edges, corners, and curves.
   We used two of these.

2. **Max Pooling Layers (The "Shrinkers")**

   After scanning, these layers shrink the picture, keeping only the most important and obvious patterns.
   This makes the model faster and more efficient.

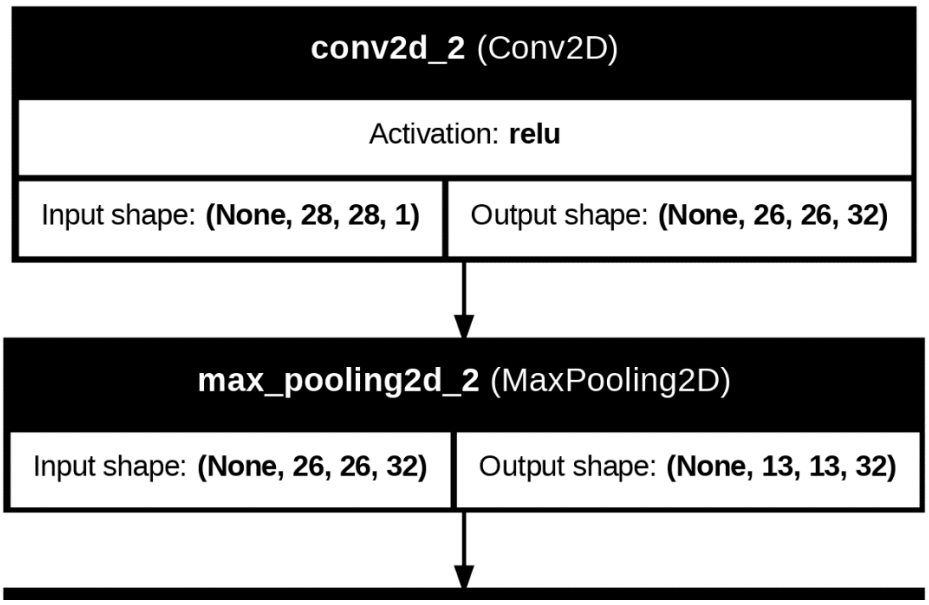
3. **Flatten Layer (The "Flattener")**

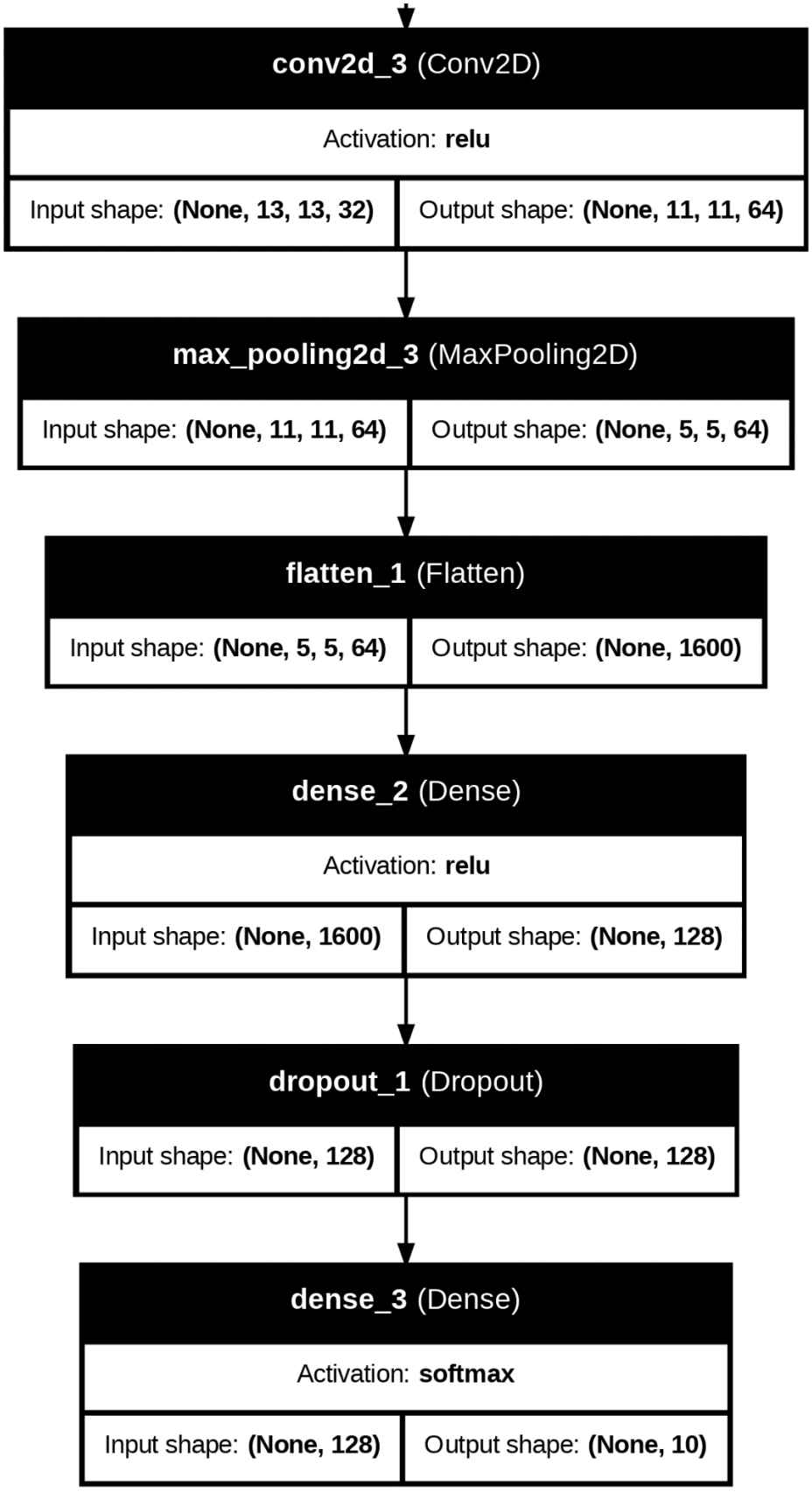
   This layer takes the 2D, shrunken-down patterns and flattens them into one long, single line of numbers.

4. **Dense Layers (The "Decision Makers")**

   This is the "brain" part that looks at the long line of numbers and makes a final decision.
   The last layer has **10 neurons**, one for each type of clothing.
   The neuron with the highest score is the model's final guess.

This diagram shows the exact blueprint of the model we built:

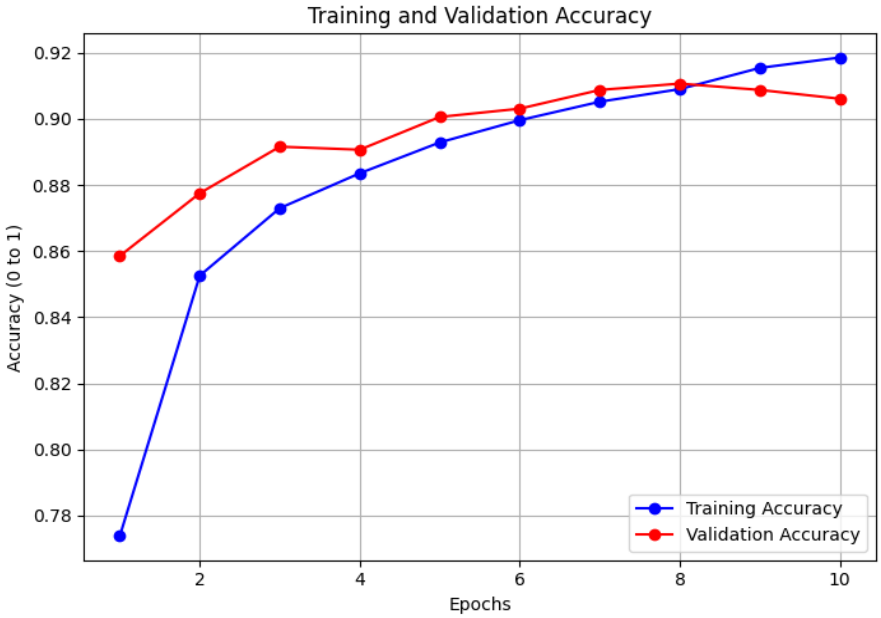


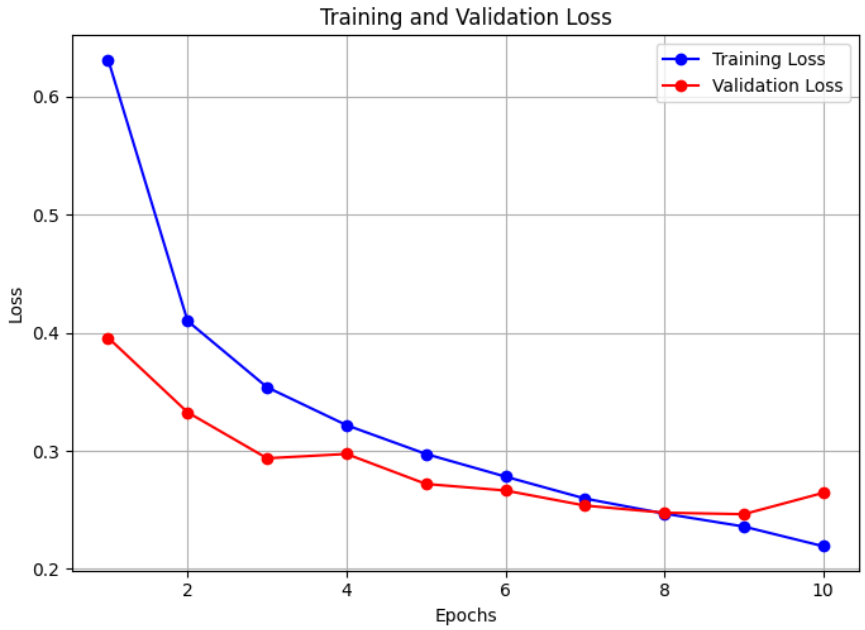

conv2d_3 (Conv2D)
Activation: relu
Input shape: (None, 13, 13, 32)
Output shape: (None, 11, 11, 64)
max_pooling2d_3 (MaxPooling2D)
Input shape: (None, 11, 11, 64)
Output shape: (None, 5, 5, 64)
flatten_1 (Flatten)
Input shape: (None, 5, 5, 64)
Output shape: (None, 1600)
dense_2 (Dense)
Activation: relu
Input shape: (None, 1600)
Output shape: (None, 128)
dropout_1 (Dropout)
Input shape: (None, 128)
Output shape: (None, 128)
dense_3 (Dense)
Activation: softmax
Input shape: (None, 128)
Output shape: (None, 10)

4. The Training: How Our Model Learned

We trained our model by "showing" it our training pictures **10 times** (called 10 "epochs").
We used **80% of our training data (48,000 images)** to teach the model, and the other **20% (12,000 images)** as a "practice test" to check if it was actually learning — not just memorizing.

These graphs show the model's learning progress:





Analysis (What These Graphs Mean)

Accuracy Graph:
This shows how many pictures the model guessed correctly.
You can see both the "Training" (blue line) and "Validation" (red line) scores went up — this is great!
It means the model was learning to get more answers right.
The red line (our "practice test") leveled off at around **90% accuracy**.

Loss Graph:
"Loss" is a number that measures how "wrong" the model's guesses were.
A lower number is better. You can see both lines went down, which is exactly what we want.
This means the model's guesses were getting less and less wrong over time.

5. The Final Exam: How Did Our Model Do?

After training was finished, we gave the model its **"final exam"** using the **10,000 test pictures** that it had never seen before.

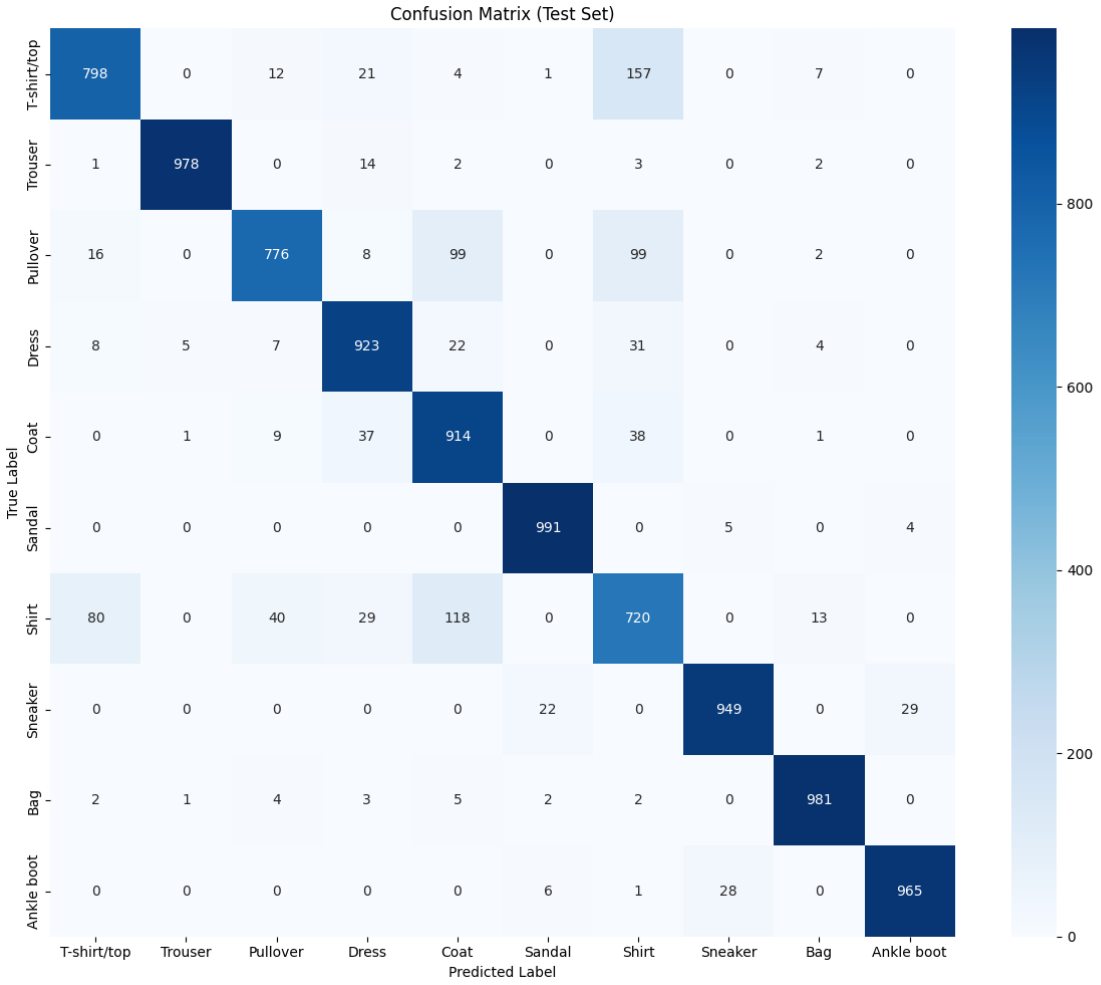
The final result was:

Final Test Accuracy: 91.07%

> This means our model correctly identified the clothing in new pictures almost **9 out of 10 times**, which is a very good result.

The "Report Card" (Confusion Matrix)

To see where the model did well and where it got confused, we made a **confusion matrix**.



Analysis (How to Read This)

- The bright, diagonal line from top-left to bottom-right shows all the **correct guesses**.
- The numbers in the other boxes are **mistakes**.

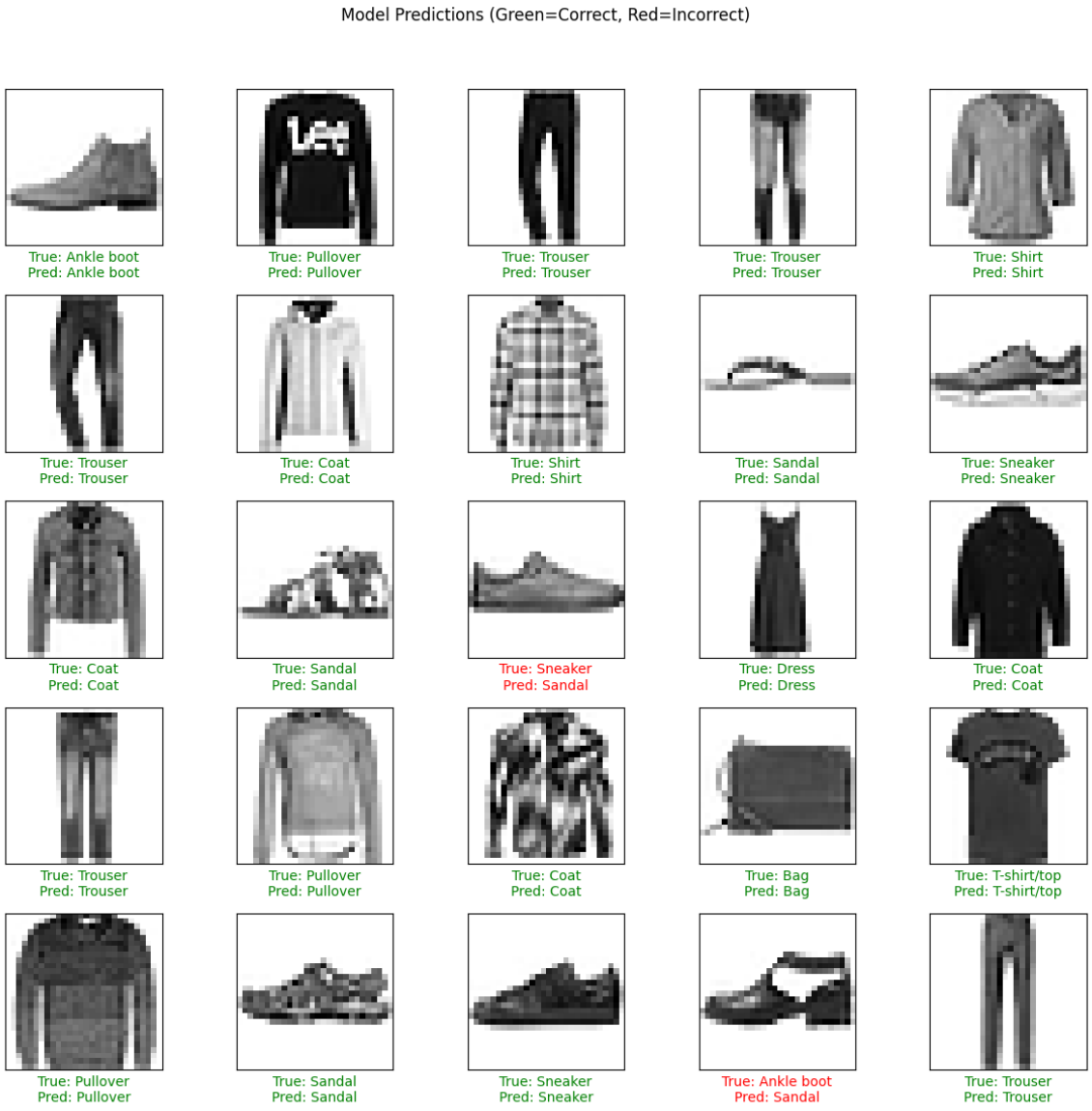
What it did well:
You can see it was great at spotting *'Trouser'* (978 correct guesses) and *'Sandal'* (991 correct guesses).

Where it got confused:
The "messiest" box is *'Shirt'*.
The model often confused *'Shirt'* with *'T-shirt/top'* and *'Pullover'*.
This makes sense because even a human would have trouble telling those apart in a small, blurry, black-and-white picture.

6. Seeing What the Model "Sees"

This last picture shows us **25 random examples** from the "final exam."

If the label is **green**, the model's guess was **correct**.
If the label is **red**, the model's guess was **wrong**.



Just like our "report card" showed, you can see the mistakes (in red) are where the model confused a *'Sandal'* for a *'Sneaker'* or a *'Sandal'* for a *'Ankle Boot'*.

7. Observations & Conclusion

This project was a success.
We proved that we can build a simple "brain" (a CNN) that can learn to "see" and classify different types of clothing with high accuracy.

The main takeaway is that **CNNs are very powerful**, but their mistakes are often logical.
The model got confused on the same things a human would get confused on.

We learned how to **build, train, and evaluate** a deep learning model from start to finish.