Automatic Clothing Classification from Images using Fashion MNIST

Language: Python | Dataset: Fashion MNIST

Students: Alikhan Bekkaliyev, Sultantore Kudaibergen, Abylai Kenesbek, Zangar Kenes

Lecturer: Maira Kopzhasarova

Discipline: AI

1. Introduction and Problem Statement

In recent years, the field of computer vision has grown rapidly thanks to advancements in deep learning techniques. One of the most practical applications of computer vision is image classification — the task of assigning a label to an image based on its visual content. In this project, we address a real-world problem from the fashion industry: automatic clothing classification from images. The ability to classify clothing items efficiently and accurately has various commercial and social implications. Online retailers, fashion assistants, recommendation systems, and accessibility tools for visually impaired users all benefit from such technology.

Our goal is to develop a deep learning-based system capable of classifying clothing items from photographs using the Fashion MNIST dataset. We aim to implement a convolutional neural network (CNN) that can learn visual patterns and generalize well to unseen examples. The solution should achieve high accuracy and robustness while remaining computationally efficient.

This project covers every major phase of an AI pipeline: problem formulation, dataset acquisition and preprocessing, model design, training and evaluation, ethical analysis, and documentation. We also take into account important concerns such as fairness, bias, and privacy, which are crucial in the deployment of any AI model.

2. Dataset and Preprocessing

The dataset used in this project is Fashion MNIST, a well-known benchmark provided by Zalando Research. It contains a total of 70,000 grayscale images of size 28x28 pixels. The dataset includes 10 classes of clothing items, each represented by a numerical label (0–9). The classes are:

0 – T-shirt/top

1 – Trouser

2 – Pullover

3 – Dress

4 – Coat

5 – Sandal

6 – Shirt

7 – Sneaker

8 – Bag

9 – Ankle boot

The dataset is split into 60,000 training images and 10,000 testing images. Each image represents a single item of clothing, centered and scaled to fit within the image frame.

Before feeding the data into the model, several preprocessing steps were carried out:

• Normalization: Pixel values were originally in the range [0, 255]. We scaled them to [0, 1] by dividing by 255.

• Reshaping: The images were reshaped to (28, 28, 1) to make them compatible with convolutional layers.

• One-hot encoding: Class labels were converted to one-hot vectors.

• Data augmentation: Random rotations, shifts, and flips were applied during training to improve generalization.

We also used Matplotlib to visualize class distribution and inspect samples, confirming the dataset is well-balanced.

3. Model Implementation

To classify the clothing items, we implemented a convolutional neural network (CNN) using the TensorFlow/Keras framework. CNNs are effective for image classification because they can capture spatial hierarchies by learning local patterns.

Model architecture:

• Input layer (28x28x1)

• Conv2D (32 filters, 3x3, ReLU)

• MaxPooling2D (2x2)

• Conv2D (64 filters, 3x3, ReLU)

• MaxPooling2D (2x2)

• Dropout (0.25)

• Flatten

• Dense (128 units, ReLU)

• Dropout (0.5)

• Dense (10 units, softmax)

Compiled with:

• Optimizer: Adam

• Loss: Categorical Crossentropy

• Metrics: Accuracy

The model was trained for 15 epochs with batch size 128. Data augmentation helped reduce overfitting and improve generalization.

4. Model Evaluation

After training, we evaluated the model on the test set (10,000 images). The CNN achieved a test accuracy of approximately 91%.

Metrics used:

• Accuracy: Proportion of correct predictions

• Confusion matrix: Helped identify confusion between visually similar classes like T-shirt and Shirt

• Precision, Recall, F1-score: Detailed per-class performance

• Accuracy/Loss plots: Showed steady convergence and minimal overfitting

The results indicate strong performance and robustness, suitable for real-world applications.

5. Ethical Considerations

We considered multiple ethical dimensions in developing the system:

• Bias: The dataset is balanced, but real-world datasets may have demographic or cultural bias.

• Privacy: Clothing classifiers in real-world use might analyze personal photos; privacy protections are essential.

• Accessibility: This technology can aid visually impaired users in identifying clothes.

• Transparency: We visualized learned filters and activations to aid interpretability.

• Environmental impact: Efficient model training with early stopping helps reduce carbon footprint.

Responsible AI development includes fairness, security, and societal value.

6. Tools and Libraries Used

We used Python and the following libraries:

• NumPy – Numerical operations

• Pandas – Basic data inspection

• Matplotlib / Seaborn – Visualization

• TensorFlow / Keras – Deep learning

• Scikit-learn – Evaluation metrics

• Google Colab – Development and GPU training environment

All code was organized, documented, and version-controlled using GitHub.

7. Results and Conclusion

The model achieved strong performance in classifying clothing items from the Fashion MNIST dataset. Major findings include:

• Over 91% test accuracy with basic CNN architecture

• Augmentation and dropout effectively reduce overfitting

• Ambiguities between some classes are a challenge

• Evaluation shows consistent results across most classes

This system could be extended with color images, larger datasets, or transfer learning. Potential real-world applications include fashion assistants, automated product tagging, and accessibility tools.

8. References

1. Fashion MNIST Dataset – https://github.com/zalandoresearch/fashion-mnist

2. TensorFlow – https://www.tensorflow.org/

3. Keras – https://keras.io/

4. Scikit-learn – https://scikit-learn.org/

5. Goodfellow et al., Deep Learning, MIT Press

6. Chollet F., Deep Learning with Python, Manning