Exploring Fashion MNIST Dataset with Deep Learning and Machine Learning

1. Introduction – Project Aim

We worked on the Fashion MNIST image dataset, which is available within the Keras library, using deep learning and machine learning algorithms. We then evaluated the effectiveness of our model using evaluation metrics.

2. Material - Method

Evaluation Metrics

- **Accuracy**: Ratio of correctly classified samples to total samples
- **Precision**: Ratio of true positive results among predicted positives, with weighted average
- **Recall**: Ratio of true positive results among actual positives
- **F1 Score**: Harmonic mean of precision and recall, balancing the two

Image Dataset

The image dataset used was the Fashion MNIST dataset, available in the Keras library.

1

2

3

Algorithms

We used two algorithms:

Deep Learning

Artificial Neural Network (ANN) using TensorFlow's Keras API

Machine Learning

K-Nearest Neighbors (KNN) algorithm, a supervised learning method

3. Information about the Fashion Mnist dataset

Image Dimensions

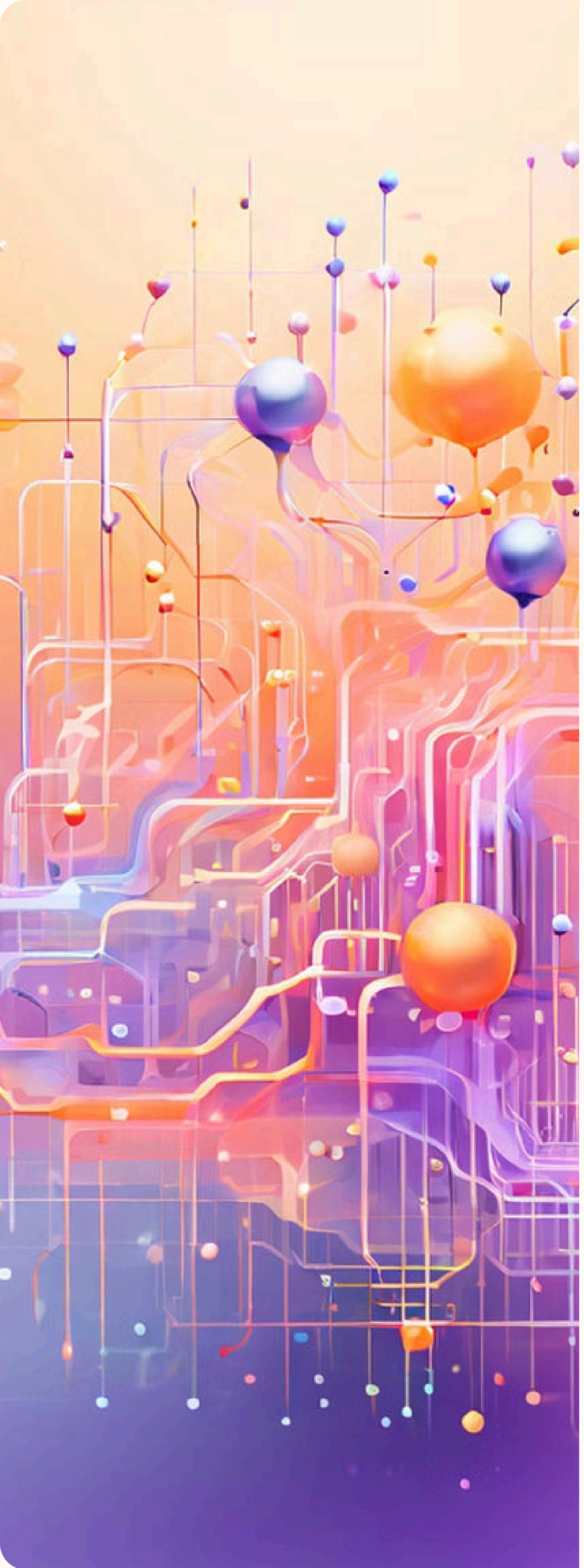
Each image in the Fashion MNIST dataset is 28x28 pixels in size, with pixel values ranging from 0 to 255 on a grayscale spectrum.

Data Quantity

The dataset contains a total of 70,000 examples, with 60,000 images in the training set and 10,000 in the test set.

Class Labels

There are 10 different clothing categories, including T-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot.



4. Models That Used:

a. Artificial Neural Network (ANN) Model

1. **Input Layer**

   The input layer flattens the 28x28 pixel images into a 1D vector of 784 elements, making them suitable for further processing by the neural network.

2. **Hidden Layer**

   The hidden layer consists of 128 neurons with a ReLU activation function, which helps the model learn and map abstract features to a higher-dimensional space.

3. **Output Layer**

   The output layer has 10 neurons, corresponding to the 10 clothing categories, with a Softmax activation function to produce a probability distribution over the classes.

b. K-Nearest Neighbors (KNN) Model

Simplicity

The KNN algorithm is relatively simple to implement and understand, as it does not require an explicit training phase and can be applied directly to the dataset.

Prediction

For each test sample, KNN identifies the 3 closest training samples and assigns the most common class among them as the predicted class.

Limitations

KNN can be computationally intensive for large datasets and its performance can degrade with high-dimensional data due to the curse of dimensionality.

c. Evaluation Metrics

1. **Accuracy**

   Measures the ratio of correctly classified samples to the total samples.

2. **Precision**

   Evaluates the ratio of true positive results among the results predicted as positive by the model.

3. **Recall**

   Evaluates the ratio of true positive results among the actual positive samples.

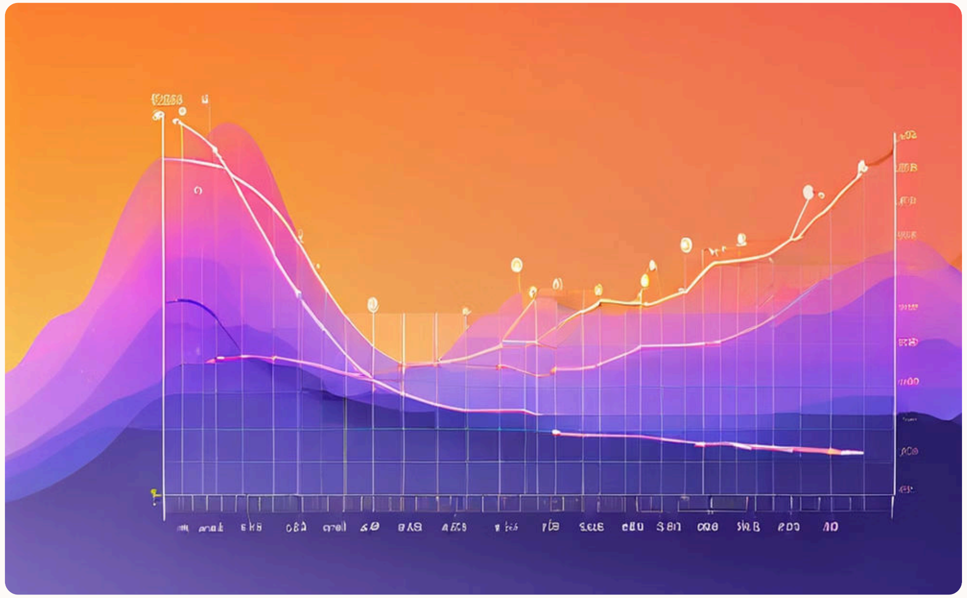
4. **F1 Score**

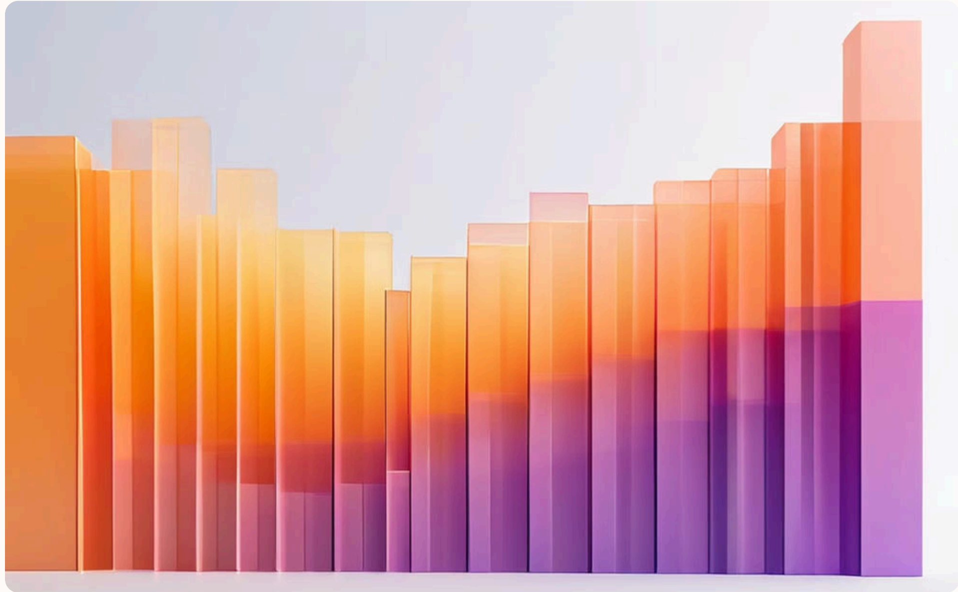
   Provides a balance between precision and recall, as the harmonic mean of the two metrics.

5. Experiment Results

	ANN Model Results	KNN Model Results
Accuracy	0.8791	0.8541
Precision	0.8812740257230547	0.8575414622679564
Recall	0.8791	0.8541
F1 Score	0.8795203780474612	0.8539002124666113

6. Argument
a. For Experiement Results:





ANN Model Performance

The ANN model achieved an accuracy of 88.93% on the test set, along with high precision, recall, and F1 score, demonstrating its effectiveness in classifying the Fashion MNIST images.

Comparison with KNN

While the KNN model also performed well, with an accuracy of 85.41%, the ANN model outperformed it across all evaluation metrics, showcasing the advantages of deep learning for complex image classification tasks.

b. Advantages and Disadvantages

Complex Pattern Learning

ANN models can learn and represent complex patterns and relationships in the data, leading to superior performance in image classification.

Computational Intensity

Training ANN models can be computationally intensive and time-consuming, especially for large datasets, compared to simpler algorithms like KNN.

Hyperparameter Tuning

ANN models require careful tuning of hyperparameters, such as learning rate and network architecture, to achieve optimal performance.

Simplicity of KNN

The KNN algorithm is easy to implement and understand, with no explicit training phase, making it a more straightforward choice for some applications.

c. Conclusion

1

Dataset Exploration

The Fashion MNIST dataset provided a valuable opportunity to explore and compare the performance of deep learning and machine learning algorithms in image classification.

2

Model Comparison

The ANN model demonstrated superior performance compared to the KNN model, highlighting the advantages of deep learning for complex tasks, despite its higher computational requirements.

3

Future Considerations

The insights gained from this project can inform future decisions on algorithm selection and guide further exploration of deep learning techniques for image classification and beyond.

7. References

TensorFlow Documentation

Scikit-learn Documentation

Keras Documentation

Fashion Mnist Dataset

Wikipedia