IMAGE CLASSIFICATION

Using Deep Learning

Presented By:
Anjali
2019532
Under the mentorship of:
Mr. Ashwini Kumar,
Assistant Professor

Table Of Content

Introduction to Image Classification

What constitutes an Image?

Images are composed of pixels, serving as the fundamental digital representation. Each pixel encapsulates details such as color, shading, or opacity. These representations are commonly expressed in three primary formats:

1. Grayscale
2. RGB (Red, Green, Blue)
3. RGBA (Red, Green, Blue, Alpha)

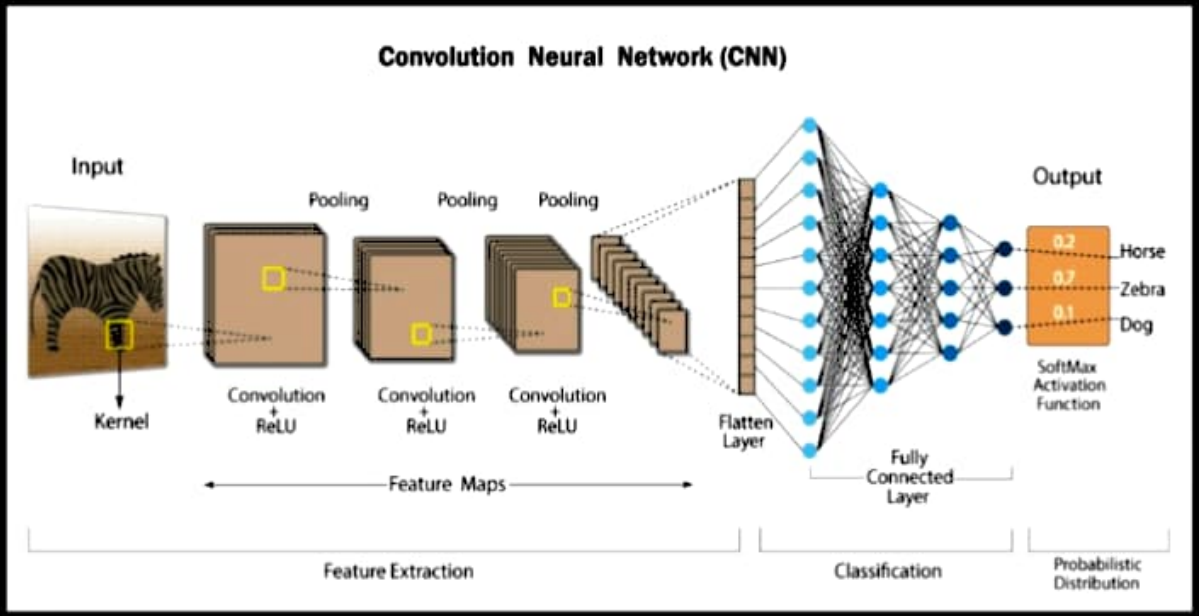
Image classification is the process of sorting images into predetermined classes or categories. This task primarily focuses on training a model to differentiate and recognize diverse objects, scenes, or unique patterns depicted within images. The fundamental aim is to equip the model with the ability to precisely allocate a particular label or category to each encountered image.

It is a subset of machine learning, which involves training artificial neural networks with multiple layers to learn and predict from complex data.

In deep learning, neural networks with multiple hidden layers are designed to automatically learn hierarchical data representation.

Deep Learning

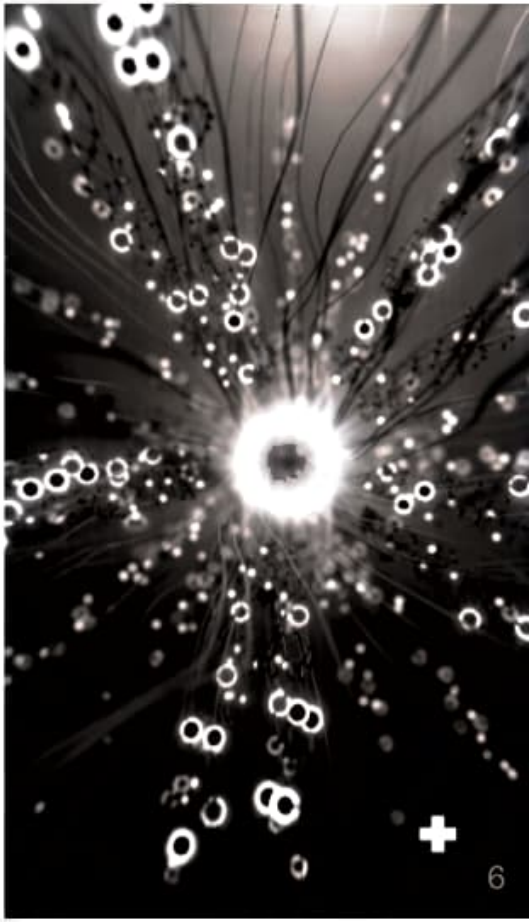
METHODOLOGY



Dataset and Model Architecture

- Fashion MNIST
- 60,000 training images, 10,000 test images.
- 28x28 grayscale images categorized into 10 classes
- CLASSES- T-shirts, trousers, pullovers, dresses, coats, sandals, shirts, sneakers, bags, ankle boots.

- CNN layers for feature extraction.
- Max-pooling for down sampling.
- Dense layers for classification.
- ReLU activation for handling pixel values.
- Softmax for final classification.
- Dropout layer to combat overfitting.
- Batch Normalization for input



The Programming Approach

- Language used - Python
- Libraries used - pandas, numpy, tkinter, matplotlib, Keras.
- Steps - Data reading, exploratory data analysis, preprocessing, Keras model construction, evaluation visualization, result prediction, deployment.
- Code execution in Google Colab or Jupyter Notebook.
- GUI development in Visual Studio Code or Jupyter Notebook.
- Trained model stored as a JSON file.

FASHION MNIST DATASET

- 70,000 grayscale images, 28x28 pixels, depicting clothing items in ten classes.
- Training Set (60,000 images) for model training, Test Set (10,000 images) for unbiased evaluation.
- Universal Benchmark: Fashion MNIST is widely adopted for refining and validating machine learning and deep learning models.
- Significance: Its simplicity and focus on clothing item recognition make it a fundamental asset in advancing image classification methodologies.

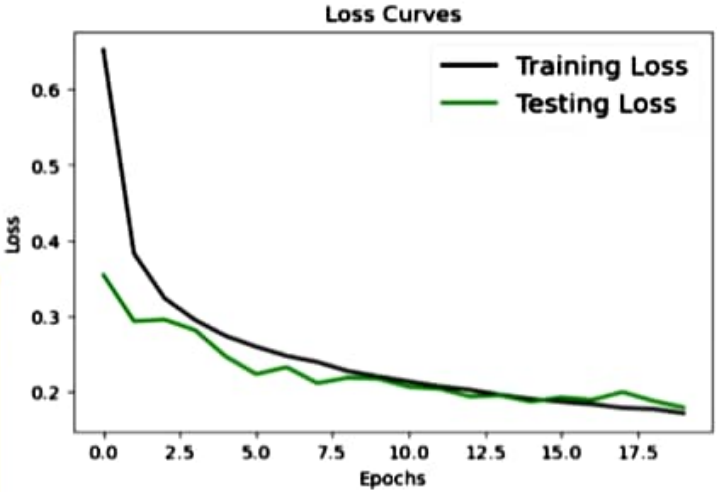
RESULT & DISCUSSION

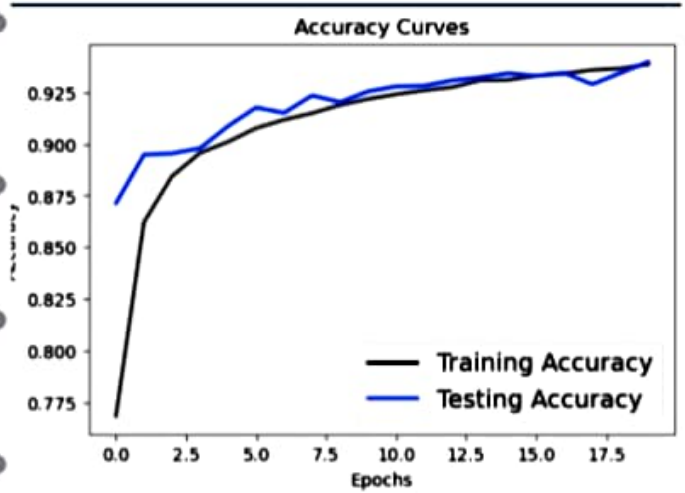
Loss Curve - Comparing the Training Loss with the Testing Loss over increasing epochs.

Accuracy Curve - Comparing the Training Accuracy with the Testing Accuracy over increasing epochs.

Model Performance:

- 436,458 trainable parameters.
- Training accuracy: 93.84%.
- Test accuracy: 93.97%.

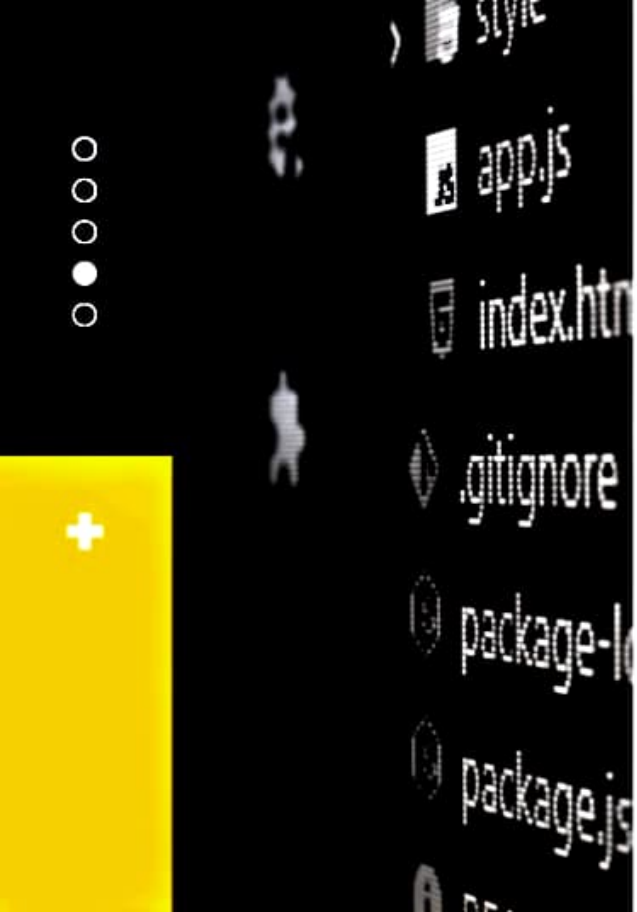

Loss Curves
Training Loss
Testing Loss
Loss
0.6
0.5
0.4
0.3
0.2
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epochs


Accuracy Curves
0.925
0.900
0.875
0.850
0.825
0.800
0.775
Training Accuracy
Testing Accuracy
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epochs

CONCLUSION

- Deep learning on datasets like Fashion MNIST is impactful for diverse industries.
- CNNs show remarkable capabilities in categorizing intricate visual patterns.
- Although our model for the Image Classification is working fine for the given data but there may have situations which when arise will lead to the wrong result production.

- Limited dataset diversity may hinder generalization to real-world scenarios.
- Low-resolution images affect intricate detail classification.
- Class imbalance leads to biased predictions and impacts model performance.
- Overfitting risks may affect adaptability to new, unseen data.
- Challenges in fine-grained classification for visually similar categories.
- Domain adaptability may require extensive retraining.



FUTURE WORK

- Develop sophisticated fashion recommendation systems.
- Extend model applicability with real-world dataset adaptation.
- Integrate into ensemble models for improved performance.
- Create personalized styling apps based on user preferences.
- Automate retail inventory management using the model.
- Incorporate into visual search technologies for fashion item exploration.


THANK YOU