6FTC2085-0901-Advanced Artificial Intelligence

Computer Vision Mini Project

Omar Faisal Amr Mourad - 21026670



Problem Statement:

This project primarily focuses on building an object recognition system using the **CIFAR-10** dataset, which contains 60,000 32x32 color images in 10 classes, such as airplanes, cars, and dogs. However, to evaluate the model more on a wider range of images, the **Fashion-MNIST** dataset was also included. Fashion-MNIST consists of 60,000 28x28 grayscale images in 10 classes of clothing items, such as T-shirts, trousers, and sneakers.

By combining these two datasets, the model is trained to recognize a total of 20 classes—10 from CIFAR-10 and 10 from Fashion-MNIST. The goal is to build an object recognition model that can accurately classify objects from both datasets and evaluate its performance on unseen data and make predictions.

Approach:

The object recognition system is built using a **Convolutional Neural Network (CNN)**

Data Preprocessing:

1. **Loading and Preprocessing the Datasets**:
    - The **CIFAR-10** and **Fashion-MNIST** datasets are loaded. CIFAR-10 contains 60,000 32x32 color images, while Fashion-MNIST contains 60,000 28x28 grayscale images. Fashion-MNIST images are resized to 32x32 pixels and converted to RGB format to match CIFAR-10's color images, histogram equalization was also applied to them to improve their contrast.
2. **Combining Datasets**:
    - The CIFAR-10 and Fashion-MNIST datasets are merged into a single training and testing set. The Training set was later on splitted into 2 sets one for training and the other for validation to tune hyperparameters. This is to ensure the model is evaluated on unseen data. After both datasets were combined they were normalized (keeping values between 0 and 1) by dividing by 255.

3. **One-Hot Encoding**:
    - The labels for both datasets are converted into one-hot encoded vectors. This is necessary for multi-class classification, where each label is represented as a binary vector, with a 1 in the position of the correct class.

Model Architecture:

1. **Convolutional Layers**:
    - The model 11 convolutional layers to learn features such as edges and textures. These layers use 32 filters, then 64, and finally 128 filters in deeper layers, with 3x3 kernels to capture small patterns.
2. **Batch Normalization**:
    - Batch normalization is applied after each convolutional layer to stabilize training by normalizing the layer's outputs.
3. **Max Pooling**:
    - Max pooling is used to reduce the dimensions of the image and keep only the most important features, helping the model focus on significant information.
4. **Dropout**:
    - Dropout layers are added to reduce overfitting. During training, these layers randomly disable a fraction of neurons, preventing the model from becoming too dependent on any single feature.
5. **Dense Layers**:
    - After flattening the features from the convolutional layers, two fully connected (dense) layers are added. The output layer uses the softmax activation function to produce class probabilities.

Data Augmentation:

- To help the model generalize better, **data augmentation** techniques are used during training. These include:
    - **Width and height shift**: Random shifts of the image along the horizontal and vertical axes.
    - **Horizontal flip**: Random flipping of images to create more training examples.

Training:

- **Model Training**:
    - The model is trained using the **Adam optimizer**, which adapts the learning rate for faster and more stable training. The loss function used is **categorical cross-entropy**, ideal for multi-class classification tasks. The model is trained for **30 epochs** with a batch size of 32. Validation data is used to check performance after each epoch.

Evaluation:

1. **Performance Metrics**:
    - After training, the model is evaluated on the test set, and its performance is measured using **accuracy**, **precision**, and **recall**. These metrics show how well the model predicts the correct class and avoids misclassifications.
2. **Classification Report**:
    - A classification report is generated, which provides detailed performance metrics for each of the 20 classes in the combined dataset, helping to identify which classes the model performs well on and where it needs improvement.

Performance on Unseen Data:

The trained model is evaluated on a test set containing both CIFAR-10 and Fashion-MNIST images that were not seen during training. The classification report provides insights into the model's ability to generalize to unseen data. The report includes metrics such as:

- **Accuracy:** The overall percentage of correct predictions.
- **Precision:** The ability of the model to avoid false positives.
- **Recall:** The model's ability to identify all relevant instances for each class.

```
625/625 ━━━━━━━━━━━━━━━━━━━━ 8s 13ms/step
```

	precision	recall	f1-score	support
airplane	0.82	0.85	0.84	1000
automobile	0.92	0.94	0.93	1000
bird	0.82	0.71	0.76	1000
cat	0.76	0.61	0.68	1000
deer	0.80	0.85	0.82	1000
dog	0.82	0.70	0.76	1000
frog	0.73	0.94	0.82	1000
horse	0.89	0.85	0.87	1000
ship	0.91	0.90	0.90	1000
truck	0.86	0.94	0.90	1000
t-shirt	0.85	0.88	0.87	1000
trouser	1.00	0.97	0.99	1000
pullover	0.92	0.83	0.88	1000
dress	0.91	0.88	0.90	1000
coat	0.74	0.96	0.84	1000
sandal	0.98	0.97	0.98	1000
shirt	0.81	0.65	0.72	1000
sneaker	0.95	0.97	0.96	1000
bag	0.98	0.99	0.98	1000
ankle boot	0.96	0.96	0.96	1000
accuracy			0.87	20000
macro avg	0.87	0.87	0.87	20000
weighted avg	0.87	0.87	0.87	20000

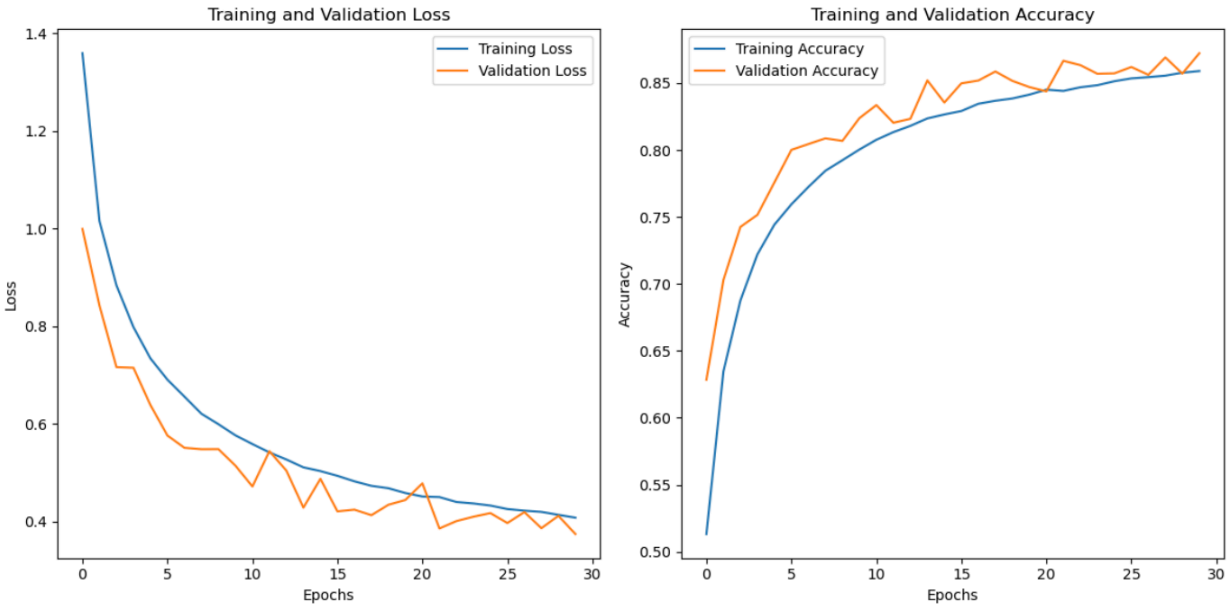
From the results, the model shows good performance in identifying objects from both datasets, with some classes showing higher accuracy than others.

Discussion of Results:

The model successfully classifies images from both CIFAR-10 and Fashion-MNIST datasets. It achieved an accuracy of 87% on the training dataset and most classes got a good precision but around 4 classes got lower precision scores than others which could lead to difficulty in predicting them.

To further assess the model's performance on individual images, we performed **prediction tests** on selected images from the test set. For each image, the model was used to predict the class label, and the true label was compared to the predicted label.

In most cases, the model correctly identified the object, with the predicted label matching the true label. However, there were some misclassifications, particularly when visually similar objects from the two datasets were involved.



The graphs show a strong learning trend for both training and validation sets. In the **training loss graph**, the blue line indicates that the model's loss consistently decreases, suggesting effective learning from the training data. This decrease is a positive sign that the model is successfully minimizing its error during training. The **validation loss** (represented by the orange line) also decreases, though with some fluctuations early on. This trend shows that the model is generalizing well to unseen data, with the validation set supporting the conclusion that the model is not overfitting to the training data.

In terms of accuracy, both the **training accuracy** and **validation accuracy** improve steadily over time. The blue training accuracy line shows a rise, reaching around 0.85, while the orange validation accuracy approaches 0.80, indicating good generalization. The small gap between training and validation accuracy suggests that the model is not overfitting but could benefit from further optimization to improve validation performance. While the model shows strong performance overall, some fine-tuning, such as adjusting the learning rate or adding more advanced techniques, could help reduce the gap and enhance its generalization ability further.

In conclusion, the object recognition system demonstrated solid performance on the CIFAR-10 and Fashion-MNIST datasets, providing a good foundation for future advancements in object recognition tasks. The prediction testing helped identify specific cases where the model could improve, guiding future refinements.