Report On Image Recognition Using Convolutional Neural Networks (CNNs)

Objective

The objective of this project was to design, train, and evaluate an image recognition model using Convolutional Neural Networks (CNNs) to classify images from the CIFAR-10 dataset. CIFAR-10 consists of 60,000 32x32 color images across 10 predefined classes (such as airplanes, dogs, cats, etc.).

Approach:

1. Dataset:

- **CIFAR-10**: The dataset contains 60,000 32x32 color images in 10 classes. The dataset is divided into 50,000 training images and 10,000 test images. Each image is represented as a 3-dimensional array (32x32x3), where 3 represents the RGB channels.

2. Preprocessing:

- **Normalization**: Each pixel value of the image was normalized to a range of [0, 1] by dividing by 255.
- **Data Augmentation**: To improve the model's generalization, data augmentation techniques such as random horizontal flipping and rotation were applied to the training images.
- **Train-Test Split**: The dataset was already split into a training set (50,000 images) and a test set (10,000 images), so no further split was necessary.

3. CNN Architecture:

The CNN architecture was designed with the following layers:

- **Convolutional Layers**: These layers perform convolution operations to learn feature maps, which capture spatial hierarchies.
  - First layer: 32 filters with 3x3 kernel size, followed by ReLU activation.
  - Second layer: 64 filters with 3x3 kernel size, followed by ReLU activation.
  - Third layer: 128 filters with 3x3 kernel size, followed by ReLU activation.

- **Pooling Layers**: Max-pooling layers were used after each convolutional block to down sample the feature maps, reducing their spatial dimensions.
- **Fully Connected Layers**: After the convolutional and pooling layers, the output is flattened and passed through fully connected (dense) layers.
    - One hidden layer with 512 units and ReLU activation.
    - Output layer with 10 units (one for each class) and softmax activation to predict the class probabilities.
- **Dropout**: Dropout regularization was applied to the fully connected layer to avoid overfitting.

4. Model Training:

- **Loss Function**: Categorical Cross-Entropy was used since the problem is a multi-class classification task.
- **Optimizer**: Adam optimizer was used due to its efficiency in handling sparse gradients and its adaptive learning rate.
- **Batch Size**: A batch size of 64 was used.
- **Epochs**: The model was trained for 20 epochs to allow sufficient time for convergence.

5. Evaluation Metrics:

- **Accuracy**: The primary metric for evaluating model performance.
- **Confusion Matrix**: Used to assess the model's performance on each individual class, providing insights into misclassifications.
- **Precision, Recall, F1-Score**: Calculated for each class to evaluate the performance of the model beyond just accuracy, especially useful for imbalanced datasets.

Results

1. **Model Accuracy**: The model achieved an accuracy of approximately **85-88%** on the test set after 20 epochs. This indicates the model's strong performance, given the complexity of the CIFAR-10 dataset.
2. **Confusion Matrix**: The confusion matrix helped in identifying which classes were frequently misclassified. For instance, classes like "airplane" and "automobile" were often confused, likely due to the similarity in shape and size between these objects.

3. **Precision, Recall, F1-Score**:
    - Precision and recall values were calculated for each class, with the model performing well on most classes.
    - Some classes, such as "cat" and "dog," had a lower F1-score due to similarities in texture and appearance, which made them harder to distinguish.

Analysis

Strengths:

- Effectiveness: The CNN architecture with multiple convolutional and pooling layers proved effective in learning spatial hierarchies from the images.
- Data Augmentation: The application of data augmentation helped the model generalize better and prevented overfitting, which is common when training on relatively small image datasets like CIFAR-10.
- Optimization: The Adam optimizer, combined with categorical cross-entropy loss, allowed for efficient training and quick convergence.

Weaknesses:

- Class Imbalance: While the accuracy was generally high, the model still struggled with some classes. This is indicative of the inherent challenges with the CIFAR-10 dataset, where certain classes have more complex visual features and thus more difficulty being classified.
- Overfitting Potential: Despite data augmentation, the model still has a potential for overfitting, especially if the number of epochs is increased significantly. Techniques like dropout and batch normalization can help mitigate this.

Conclusion

The CNN model performed well on the CIFAR-10 dataset, achieving an accuracy of around 85-88%. The model's ability to classify images into 10 categories shows that CNNs are highly effective for image recognition tasks. However, there is room for further improvement, especially for harder-to-classify classes. Future improvements can involve more advanced techniques like transfer learning or deeper architectures.