**Assignment No: - 3**

**Image Classification using CNNs**

**Problem Statement:**

Implement Image classification using convolutional neural networks (CNNs) for multiclass

classification.

**Objectives:**

1. To implement image classification using Convolutional Neural Networks (CNNs) for multiclass classification.
2. To understand the architecture of CNNs and how they are applied to extract features from images for classification tasks.
3. To explore the practical applications of CNNs in real-world scenarios such as object detection, image recognition, and classification.

**Theory:**

**Image classification** is the task of assigning a label to an image from a predefined set of classes. Convolutional Neural Networks (CNNs) are a type of deep learning model specifically designed for working with grid-like data such as images. They consist of convolutional layers that automatically extract spatial features from the input images, followed by fully connected layers for classification.

CNNs are particularly effective in image-related tasks due to their ability to capture spatial hierarchies of patterns (edges, textures, objects) through their layered structure. For multiclass classification, CNNs are used to assign a single label from multiple categories to each image.

**Methodology:**

1. **Data Collection**: Obtain a labeled dataset of images for multiple classes (e.g., CIFAR-10, MNIST, or custom datasets).
2. **Data Preprocessing**:
   * Resize images to a fixed size for input into the CNN.
   * Normalize pixel values (e.g., scaling between 0 and 1).
   * Split the data into training, validation, and testing sets.
3. **Model Design**:
   * **Convolutional Layers**: Apply filters to the input image to extract spatial features such as edges and textures.
   * **Pooling Layers**: Reduce the spatial dimensions of the feature maps, preserving the most important information while reducing computation.
   * **Fully Connected Layers**: Connect all neurons from the previous layers to the output, mapping the extracted features to class labels.
   * **Activation Functions**: Use ReLU for non-linearity in hidden layers and softmax for the final output layer for multiclass classification.
4. **Model Training**:
   * Use **cross-entropy loss** to measure the error between predicted and actual labels.
   * Apply **backpropagation** and an optimizer (e.g., Adam) to minimize the loss by updating the weights.
5. **Evaluation**: After training the model, evaluate its performance on the test set by measuring accuracy, precision, recall, and F1-score.
6. **Deployment**: Once validated, deploy the model for real-world use cases such as image recognition applications.

**Working Principle / Algorithm:**

1. **Convolutional Layers**:
   * The input image is passed through multiple convolutional layers where small filters (kernels) scan the image to detect local patterns.
   * The output of each convolution is a feature map, capturing the presence of specific features at different positions in the image.
2. **Pooling Layers**:
   * Pooling layers reduce the dimensions of the feature maps, preserving the most important information while reducing computation.
   * Typically, max-pooling is used to retain the maximum value in each region.
3. **Fully Connected Layers**:
   * After multiple convolution and pooling operations, the feature maps are flattened into a single vector.
   * This vector is passed through fully connected layers to perform classification.
4. **Softmax Function**:
   * The output layer uses the softmax function to convert the raw scores into probabilities for each class.
   * The class with the highest probability is chosen as the predicted label.
5. **Backpropagation**:
   * The error is computed using the cross-entropy loss function and backpropagated through the network to adjust the weights, minimizing the error.

**Advantages:**

1. **Automatic Feature Extraction**: CNNs automatically learn relevant features from images, reducing the need for manual feature engineering.
2. **High Accuracy**: CNNs are known to provide excellent performance on image classification tasks, especially when trained on large datasets.
3. **Spatial Hierarchy**: CNNs capture hierarchical structures in images (from low-level edges to high-level objects) due to their layered design.
4. **Wide Applicability**: CNNs are widely used in various fields such as medical imaging, autonomous vehicles, and object recognition.

**Disadvantages / Limitations:**

1. **Requires Large Datasets**: CNNs perform best with large amounts of labeled data, which may not always be available.
2. **Computationally Intensive**: CNNs require significant computational power and memory, especially for large images and deep networks.
3. **Overfitting**: CNNs can overfit on small datasets, capturing noise instead of actual patterns, leading to poor generalization on new data.
4. **Black Box**: CNNs are often seen as black boxes, making it difficult to interpret why a certain prediction was made.

**Diagram:**



**Conclusion:**

In conclusion, Convolutional Neural Networks (CNNs) are a powerful and efficient approach for image classification tasks, particularly for multiclass classification. By leveraging their ability to automatically extract features from images and learn hierarchical representations, CNNs have achieved state-of-the-art performance across various applications. However, practitioners should be mindful of the data requirements, computational costs, and the potential for overfitting. With proper management and optimization, CNNs can be effectively applied to solve complex image classification problems.