

Name: Yash Gaikwad
Roll No: 381071
PRN: 22420141

Assignment 3

Problem Statement: Implement Image classification using convolutional neural networks (CNNs) for multiclass classification.

Introduction:

Image classification is one of the most fundamental tasks in computer vision, where the goal is to assign a label to an image from a set of predefined categories. In this assignment, we focus on **multiclass classification**, where an input image can belong to one of multiple possible classes (e.g., classifying images of cats, dogs, and birds).

Convolutional Neural Networks (CNNs) are widely used for this task because they can automatically learn hierarchical feature representations from raw image pixels. Their ability to capture spatial and structural patterns makes them far superior to traditional machine learning methods for image classification.

Objective:

The main objectives of this assignment are:

- To understand the working of Convolutional Neural Networks (CNNs).
- To implement a CNN in Python using **Keras/TensorFlow**.
- To train the CNN on a dataset containing multiple image categories.
- To evaluate the model's performance using accuracy and loss metrics.
- To demonstrate how CNNs can effectively handle multiclass classification problems.

Theory:

1 Convolutional Neural Networks (CNNs)

CNNs are a class of deep learning models designed specifically for processing grid-like data such as images. Their key components are:

1. Convolutional Layers

- Apply filters (kernels) to extract spatial features like edges, textures, and patterns.
- Preserve the spatial relationship between pixels.

2. Pooling Layers

- Reduce dimensionality by summarizing feature maps (e.g., Max Pooling).
- Help make the model more efficient and invariant to small transformations.

3. Fully Connected Layers

- Flatten the features and pass them into dense layers.

- Learn high-level combinations of extracted features for classification.

4. **Activation Functions**

- ReLU (Rectified Linear Unit) is commonly used in hidden layers for non-linearity.
- Softmax is used in the output layer for multiclass classification to assign probabilities to each class.

2 Workflow for Multiclass Image Classification

1. **Dataset Preparation**

- Collect images belonging to multiple categories.
- Preprocess images (resize, normalize pixel values, and perform augmentation).

2. **Model Building**

- Define a CNN with multiple convolutional and pooling layers.
- Use a **softmax output layer** with the number of neurons equal to the number of classes.

3. **Model Training**

- Compile the model with **categorical cross-entropy loss** and optimizers like Adam/SGD.
- Train the CNN on training data while monitoring validation accuracy.

4. **Model Evaluation**

- Evaluate using metrics such as accuracy, precision, recall, and confusion matrix.
- Visualize training/validation loss and accuracy curves.

5. **Prediction**

- For a new image, preprocess it and pass it through the trained model.
- The model outputs probabilities for each class, and the class with the highest probability is chosen as the prediction.

Conclusion:

In this assignment, we successfully implemented a **Convolutional Neural Network (CNN)** for multiclass image classification. By training the CNN on a dataset with multiple categories, we observed its ability to learn meaningful features and achieve high accuracy compared to traditional methods. This project demonstrates the effectiveness of CNNs in solving complex image-based tasks. The same approach can be extended to more advanced architectures such as VGG, ResNet, and Inception for improved performance in real-world applications.