|  |  |
| --- | --- |
| **Ex No: 3b**  **Date: 12thSeptember 2024** | **Small Image Classification Using Convolutional Neural Network** |

**Objective:** The goal of this experiment is to develop, train, and assess the performance of both Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) models for image classification using the CIFAR10 dataset from TensorFlow and Keras. This multi-class classification task (10 classes) seeks to compare the models by tuning hyperparameters, analyzing accuracy, and determining which architecture is better suited for the task.

**Description:**

**Artificial Neural Networks (ANN)** are machine learning models inspired by the structure of biological neural networks in the human brain. They consist of layers of neurons (input, hidden, and output) that transform input data through nonlinear functions, learning patterns by adjusting weights during training. ANNs are well-suited for tabular or low-dimensional data but can struggle with high-dimensional data like images.

**Convolutional Neural Networks (CNN)** a specialized type of ANN, are primarily designed for image and video processing. They use convolutional layers that apply filters (kernels) to input data to detect spatial hierarchies and features like edges, textures, and shapes. CNNs are highly effective in image classification tasks due to their ability to capture spatial dependencies through pooling layers and convolution operations.

**Model Summary:**

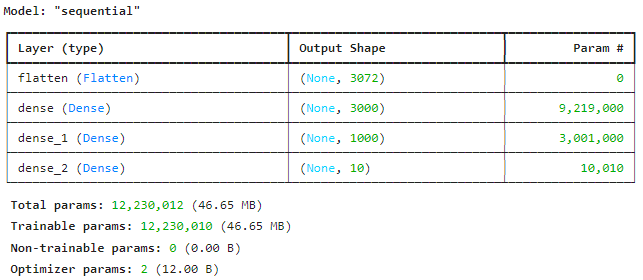
****

Figure 1: A sequential ANN model for Small Image classification.

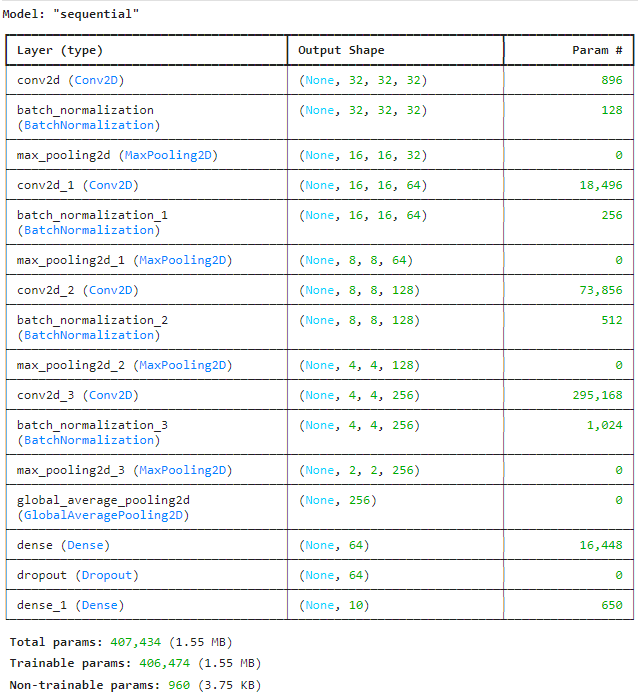


Figure 2: A sequential CNN model for Small Image classification.

**Building the parts of the algorithm**

Here are the steps involved in building each part of the algorithm:

1. **Data Preprocessing**: The dataset is loaded and divided into training and testing sets. The class labels of the input data are converted into binary class matrices for multi-class classification.

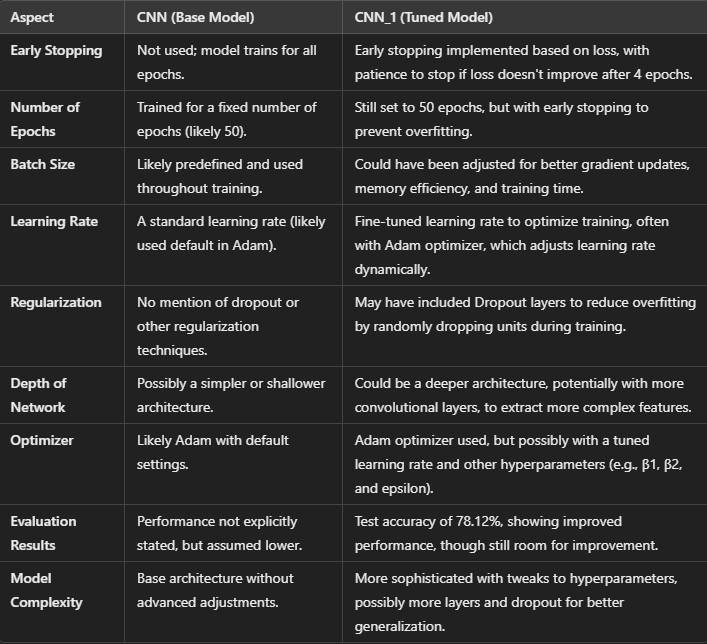
**Model Architecture**:  
The neural network models are built using Keras, comprising multiple layers such as input, hidden (dense and dropout), and output layers.

* **ANN Model**:
  + **Input Layer**: Accepts flattened image data (usually 1D).
  + **Hidden Layers**: Composed of fully connected (dense) layers with activation functions like ReLU.
  + **Output Layer**: Uses a sigmoid activation function for multi-class classification.
* **CNN Model**:
  + **Convolutional Layers**: Apply filters to perform convolutions on image data, extracting key features.
  + **Activation Function**: ReLU is used to introduce non-linearity into the model.
  + **Pooling Layers**: Perform downsampling of feature maps to reduce dimensionality and computational load.
  + **Flatten Layer**: Converts the 2D feature maps into a 1D array for the fully connected layers.
  + **Fully Connected Layers**: Dense layers at the end used for classification.
  + **Output Layer**: Uses softmax activation for multi-class classification.

1. **Compilation**:

The model is compiled with a loss function (sparse categorical cross-entropy) and an optimizer (SGD for ANN and Adam for CNN), with accuracy as the metric to track during training.

1. **Training**:
   * The model is trained over variable epochs, with accuracy and loss tracked during each epoch.



**2. Evaluation:**

- The final performance of the model is evaluated on a test set, yielding both accuracy and loss metrics.

- The model is further tested on individual test images to assess its predictions and the associated probabilities.

Inferences:

The performance of the CNN\_1 model was enhanced by implementing early stopping, which prevented overfitting and saved time by halting training when no further improvement in loss was observed. Additionally, CNN\_1 incorporated dropout regularization to further mitigate overfitting, which the base CNN model lacked. Hyperparameters, such as the number of epochs, were increased in CNN\_1. Its architecture may have been deeper or more complex than the base CNN, leading to better feature extraction and higher classification accuracy.

Conclusion:

CNN\_1 achieved a test accuracy of 78.12%, demonstrating its ability to capture significant patterns in the data, though there is still room for improvement. The classification report shows reasonable performance across various classes, but some, like class 3 (with a recall of 0.60), indicate difficulties in recognizing certain features.

Overall, the experiment concludes that CNN models outperform traditional ANN models in image classification due to their superior spatial feature extraction capabilities. However, further tuning and deeper architectures are needed for enhanced performance.

**GitHubLink:** [**https://github.com/priyankamanjunath01/deep-learning**](https://github.com/priyankamanjunath01/deep-learning)