DEEPLEARNING PROJECT REPORT

NAME : SUBHAJOY MUKHERJEE

ENROL. NO .: 12021002016029

ROLL NO.: 25

SEMISTER : 5TH

STREAM :CSE(AIML)

TITLE : IMAGE CASSIFICATION USING CNN

OBJECTIVE :

The objective of this report is to analyze and document the Convolutional Neural Network (CNN) image classification model implemented using TensorFlow and Keras. The model aims to classify images from the CIFAR-100 dataset into 100 different classes.

DATASET :

The CIFAR-100 dataset is used for training and testing the model. It consists of 60,000 32x32 color images in 100 different classes, with 600 images per class.

DATA PREPROCESSING :

1. The pixel values of the images are normalized to the range [0, 1].
2. The class labels are one-hot encoded using the `to_categorical` function.
3. Data augmentation is applied using the `ImageDataGenerator` to increase the model's robustness by introducing variations in the training data.

MODEL ARCHITECTURE :

The CNN model consists of multiple convolutional layers, activation functions, max-pooling layers, dropout layers, and fully connected layers.

1. The convolutional layers use the 'elu' (exponential linear unit) activation function.

2. Max-pooling layers with a pool size of (2, 2) are employed to reduce spatial dimensions.
3. Dropout layers with a dropout rate of 0.25 and 0.5 are used to prevent overfitting.
4. The final layer employs the softmax activation function for multi-class classification.

MODEL SUMMARY:

The model architecture is summarized as follows:

1. **Input Shape:** (32, 32, 3)
2. **Convolutional Layers:** 3 sets of (128, 3x3), (256, 3x3), (512, 3x3)
3. **MaxPooling Layers:** After each set of convolutional layers
4. **Dropout Layers:** After the second and third sets of convolutional layers
5. **Dense Layers:** Two dense layers with 1024 and the number of classes neurons, respectively.

MODEL COMPILATION :

The model is compiled using the RMSprop optimizer and categorical crossentropy loss function. Accuracy is chosen as the evaluation metric.

TRAINING:

The model is trained using the fit method with the training data generated by the ImageDataGenerator. Early stopping is implemented to prevent overfitting, with a patience of 8 epochs.

RESULTS:

1. The model's performance is evaluated on the test set, achieving a test accuracy and loss.
2. Validation loss and accuracy history plots are generated for visual analysis.
3. THE TEST ACCURACY OF THE MODEL IS : **62.45%**

VISUALIZATIONS :

The script includes visualizations of the training history, showing validation loss and accuracy over epochs.

CONCLUSIONS:

The implemented CNN model demonstrates a structured architecture for image classification on the CIFAR-100 dataset. The training history plots provide insights into the model's learning process, showing validation loss decreasing and accuracy increasing over epochs. The achieved test accuracy serves as a quantitative measure of the model's performance on unseen data. Fine-tuning and experimentation with hyperparameters may further enhance the model's accuracy and generalization capabilities.