Report: CNN Architecture and Transfer Learning for CIFAR-10 Image Classification

1. Description of the Chosen CNN Architecture:

The initial CNN model was designed using a **Sequential model** architecture. It consisted of three convolutional layers followed by max-pooling layers, batch normalization, and fully connected dense layers. The network aimed to extract relevant features from images through the following layers:

- **Convolutional Layers**: Three layers with increasing numbers of filters (32, 64, 128) and a 3x3 kernel size, using ReLU activation.
- **Max Pooling Layers**: Used after each convolutional layer to downsample the spatial dimensions of the feature maps.
- **Batch Normalization**: Implemented to improve training speed and model stability.
- **Fully Connected Layers**: A 256-unit dense layer followed by a softmax output layer for classification into 10 classes (CIFAR-10).
- **Dropout**: A 0.5 dropout was applied to the fully connected layer to prevent overfitting.

2. Explanation of Preprocessing Steps:

To improve generalization, several preprocessing and augmentation steps were implemented:

- **Gaussian Noise**: Added random Gaussian noise to the input images, helping the model become robust to noisy or imperfect data.
- **Gaussian Blur**: Applied blur to make the model more robust to input variations.
- **Image Augmentation**: Used ImageDataGenerator for real-time image augmentation:
    - **Rotation** (15 degrees), **width** and **height shifts** (10% of the image size), and **horizontal flip** were applied during training.

Data normalization was performed by dividing pixel values by 255 to scale them between 0 and 1. Training data was split into training (80%) and validation (20%) sets.

3. Details of the Training Process:

- **Optimizer**: The Adam optimizer was used with a learning rate of 0.001.
- **Batch Size**: 64 images per batch.

- **Number of Epochs**: The model was trained for 50 epochs, with early stopping based on the validation loss.
- **Early Stopping**: The training stopped early if there was no improvement in validation loss after 10 consecutive epochs, with the best weights restored.
- **Learning Rate Reduction**: If the model's performance plateaued, the learning rate was reduced by a factor of 0.5 when no improvement was observed for 5 epochs.

4. Results and Analysis of Model's Performance:

The CNN model achieved a validation accuracy of **0.7890 (78.90%)** on the CIFAR-10 validation set.

- **Classification Report**: Precision, recall, and F1-scores were generated, showing reasonable performance across most classes, with the best results in "automobile" (precision: 0.88) and "truck" (precision: 0.85).
- **Confusion Matrix**: A confusion matrix visualized model performance, revealing that some classes (like "cat" and "dog") were often misclassified due to their visual similarities.

5. Transfer Learning with VGG16 and InceptionV3:

In the next step, **transfer learning** was applied using **VGG16** and **InceptionV3**, pre-trained on ImageNet. The top layers of these models were replaced with custom layers to fine-tune them for CIFAR-10 classification.

- **VGG16**: Transfer learning using VGG16 achieved a validation accuracy of **52.3%**.
- **EfficientNet**: Transfer learning using **EfficientNet** achieved a validation accuracy of **10%**.
- **ResNet 50**: Transfer learning using **ResNet 50** achieved a validation accuracy of around **10%**.
- **InceptionV3**: Transfer learning using InceptionV3 achieved a validation accuracy of **73.10%** and a test accuracy of **72.61%**.
    - **Model Freezing**: All layers of the pre-trained network except the last 20 were frozen.
    - **Global Average Pooling**: Replaced flattening layers to reduce complexity and improve performance.

6. What is Your Best Model and Why?

Out of the transfer learning models, The **InceptionV3 transfer learning model** provided the best performance with a test accuracy of **72.61%**. This improvement can be attributed to:

- The power of transfer learning, leveraging features pre-learned from ImageNet.
- Fine-tuning only the last few layers, which allowed the model to adapt to CIFAR-10 without overfitting to the small dataset.

The best model was our CNN model with around 78%

7. Insights Gained from the Experimentation Process:

- **Importance of Preprocessing**: Image augmentations, such as Gaussian noise and blur, significantly improved model generalization.
- **Transfer Learning**: Fine-tuning models pre-trained on large datasets like ImageNet (such as InceptionV3) yielded better results than training a custom CNN from scratch.
- **Model Complexity**: While more complex models like InceptionV3 perform better, training time increases significantly, and early stopping is essential to avoid overfitting.

8. Visualizations:

- **Confusion Matrix**: Displays model performance across all classes, with most confusion observed between "cat" and "dog" classes.