Report

Approach-

Data Preprocessing: The dataset information is read from a CSV file (train.csv) containing image filenames and class labels. The data is then split into training and validation sets using train_test_split from sklearn. For image augmentation, the ImageDataGenerator is used: the training data is augmented with rescaling, rotation, width/height shift, shear, zoom, and horizontal flip to enhance variety and prevent overfitting, while the validation data is only rescaled. Image data generators are created for both training and validation sets using flow_from_dataframe, which reads filenames and class labels from the DataFrame and fetches the images from specified directories.

Model Definition: The base model used is MobileNetV2 with pre-trained ImageNet weights, excluding the top classification layers (include_top=False) to allow customization. The output of the base model is flattened, followed by a fully connected (Dense) layer with 512 neurons and ReLU activation. A Dropout layer with a 0.5 rate is added to prevent overfitting. The final layer is a Dense layer with neurons equal to the number of unique classes, using softmax activation for multi-class classification. All base model layers are frozen to keep their weights unchanged during training. The model is compiled with the Adam optimizer (learning rate 1e-4) and categorical_crossentropy as the loss function, with accuracy as the evaluation metric.

Model Training: The model is trained for 80 epochs using the fit method with training and validation data generators. The training history, which includes metrics like training and validation accuracy and loss, is stored for further analysis.

Testing and Predictions: Test image filenames are collected from the test directory and stored in a DataFrame. An ImageDataGenerator is created for the test data, applying only rescaling. The model predicts class probabilities for the test images, and predicted class labels are obtained using argmax. These predictions are saved to a CSV file (KBLD_sbmission.csv).

Accuracy and Loss-

Accuracy = 88% Loss = 0.36 Epochs = 5 Accuracy = 95.21% Loss = 0.13 Epochs = 80