***KNN Classification Model***

***Documentation: KNN Classification Project for Pet Image Dataset***

***This project demonstrates the use of k-Nearest Neighbors (KNN) classification for classifying images of pets (cats and dogs) using features extracted from a pretrained MobileNetV2 model. Below is an overview of the methods and steps used in this implementation:***

***1. Importing Required Libraries***

***The project uses several key libraries:***

* ***NumPy: For numerical operations and data manipulation.***
* ***TensorFlow/Keras: For image preprocessing and using the pretrained MobileNetV2 model.***
* ***Matplotlib & Seaborn: For visualizing results.***
* ***Scikit-learn: For implementing the KNN classifier and evaluating the model's performance.***

***2. Data Preprocessing and Augmentation***

***The images are preprocessed and augmented using the ImageDataGenerator class. This includes:***

* ***Rescaling: Pixel values are normalized to the range [0, 1].***
* ***Data Augmentation: Variations such as rotation, zoom, width/height shifts, and horizontal flips are applied to enhance the dataset.***
* ***Validation Split: The dataset is split into training and validation subsets to evaluate the model.***

***3. Loading the Data***

***The dataset is loaded from the specified directory. The ImageDataGenerator creates batches of augmented images for training and validation. The images are resized to a uniform shape of 224x224 pixels to match the input requirements of the MobileNetV2 model.***

***4. Feature Extraction Using MobileNetV2***

***A pretrained MobileNetV2 model (trained on ImageNet) is used to extract features from the images. The following steps are followed:***

1. ***The pretrained model's layers are frozen to retain their weights and prevent overfitting.***
2. ***The images are passed through the model to extract high-level feature representations (feature maps).***
3. ***The extracted features are reshaped into a flat format to be used as input for the KNN classifier.***

***5. Training the KNN Classifier***

***The extracted features are used to train a KNN Classifier. The classifier is configured to use the Euclidean distance metric to find the nearest neighbors. The number of neighbors (k) is set to 5. Once trained, the classifier is used to predict labels for the validation data.***

***6. Evaluating the Model***

***The model’s performance is evaluated using the following metrics:***

1. ***Confusion Matrix: A visualization showing correct and incorrect predictions for each class (cats and dogs).***
2. ***Classification Report: A summary that includes precision, recall, and F1-score for each class.***
3. ***Accuracy Score: A single metric representing the overall correctness of the model’s predictions.***

***7. Results and Visualizations***

* ***The Confusion Matrix highlights how well the model distinguishes between cats and dogs.***
* ***The Classification Report provides detailed insights into the model’s strengths and weaknesses for each class.***
* ***The Accuracy Score gives an overall indication of the model's effectiveness.***