Data Collection and Preprocessing Phase

Date	12 July 2024
Team ID	SWTID1720420728
Project Title	Dog Breed Identification Using Transfer Learning
Maximum Marks	6 Marks

Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	**Description**
Data Overview	The dataset consists of a diverse collection of dog images paired with corresponding breed labels sourced from various repositories. It encompasses a wide range of breeds, sizes, and characteristics necessary for comprehensive model training.
Resizing	`resize_images` function adjusts images to a desired size for preprocessing, aiding tasks like image classification, ensuring uniformity across the dataset.
Normalization	Normalizing pixel values involves adjusting them to fit within a predefined range, such as [0, 1] or [-1, 1], to enhance model training and convergence.
Data Augmentation	Augmentation techniques, like flipping, rotation, shifting, zooming, or shearing, alter training data to enhance diversity. This augmentation boosts model robustness and generalization by exposing it to variations in images, reducing overfitting, and improving performance on unseen data.
Denoising	Applying denoising filters reduces image noise, enhancing clarity and aiding in feature extraction for more accurate model predictions in dog breed identification using transfer learning.

Edge Detection	Implementing edge detection algorithms enhances feature extraction by highlighting prominent edges in images, aiding in the identification of key visual patterns essential for accurate breed classification in transfer learning models. This preprocessing step improves the model's ability to discern breed-specific characteristics from input images, leading to more robust predictions.
Color Space Conversion	Converting images from one color space to another entails changing the representation of pixel values, such as from RGB to grayscale or HSV. This process adjusts how colors are encoded, enabling different analyses or emphasizing specific image attributes for better model interpretation or performance.
Image Cropping	Cropping images involves removing unwanted parts to focus on relevant regions, enhancing model focus on key features and reducing computational load by excluding unnecessary background information.
Batch Normalization	Batch normalization stabilizes training by normalizing activations within each mini-batch, ensuring consistent mean and standard deviation. It introduces learnable parameters for optimal scaling and shifting, improving training efficiency and generalization performance.
Data Preprocessing Code Screenshots	
Loading Data	`dataset_dir = "/content/train"` `labels = pd.read_csv("/content/labels.csv")`
Resizing	`from tensorflow.keras.preprocessing.image import ImageDataGenerator # type: ignore` `train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)`

Data Augmentation	`from tensorflow.keras.preprocessing.image import ImageDataGenerator # type: ignore`
	`train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)`
	`# ***************`
	`datagen = ImageDataGenerator()`
	`generator = datagen.flow_from_directory(`
	`"/content/subset/train",`
	`target_size=(224, 224),`
	`batch_size=32,`
	`class_mode='categorical',`
	`shuffle=False`
	`)`
	`Found 10222 images belonging to 120 classes.`

Color Space Conversion