**Data Collection and Preprocessing Phase**

|  |  |
| --- | --- |
| Date | 20 October 2024 |
| Team ID | 739949 |
| Project Title | Bird Species Classification |
| 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 used is *200 Bird Species with 11,788 Images* from Kaggle, stored in Google Drive at C:/Users/manda/OneDrive/Desktop/Bird Species Classification/major/CUB\_200\_2011/CUB\_200\_2011/images. It consists of 200 bird species, each with multiple images. The images are labeled and categorized for classification. |
| Resizing | The images are resized to *(224 × 224 pixels)* to ensure uniform input size for the CNN model. The resizing is performed using torchvision.transforms.Resize() in PyTorch. Example: transforms.Resize((224, 224)). |
| Normalization | Normalization scales pixel values to a specific range for stable training. The images are normalized using [0,1] scaling (dividing by 255) or standard normalization (mean=0.5, std=0.5). The torchvision.transforms.Normalize() function is used. Example: transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5]). |
| Data Augmentation | |  | | --- | |  |  |  | | --- | | Data augmentation techniques such as random rotation, horizontal flipping, brightness adjustments, and cropping are applied to increase dataset diversity. This is done using torchvision.transforms. Example: transforms.RandomHorizontalFlip(), transforms.RandomRotation(15). | |
| Denoising | Denoising is applied using Gaussian Blurring to remove noise from images for better feature extraction. The cv2.GaussianBlur() function is used. Example: cv2.GaussianBlur(image, (5, 5), 0) |
| Edge Detection | Edge detection is performed using the Canny Edge Detection technique to highlight bird contours. The cv2.Canny() function is used. Example: cv2.Canny(image, 100, 200). |
| Color Space Conversion | The images are converted from RGB to Grayscale to highlight different features. The conversion is done using OpenCV's cv2.cvtColor() function. Example: cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY). |
| Image Cropping | Images are cropped to focus on the bird and remove unnecessary background. Bounding box cropping is applied if annotations are available. Cropping is performed using PIL or cv2. Example: cropped\_image = image[y1:y2, x1:x2]. |
| Batch Normalization | Batch normalization is applied after convolutional layers in the CNN model to stabilize training and accelerate convergence. This is done using torch.nn.BatchNorm2d(). Example: nn.BatchNorm2d(num\_features=64). |
| **Data Preprocessing Code Screenshots** | |
| Loading Data |  |
| Normalization |  |
| Data preprocessing |  |
| Pre Trained |  |
| Adding Layers |  |
| Training and Testing |  |
| Save Model |  |