Group Members: Shihua Lu, Chuhan Ren

Problem Statement:

This project aims to build an animal identification system using computer vision. Given an image containing one or more animals, the system should accurately classify the animal(s) present in the image. The input will be a color image of an animal, and the desired output is the predicted animal species (e.g., cat, dog, horse, elephant, etc.). This tool can be used in wildlife monitoring, educational tools, or pet identification applications.

Data Acquisition:

We will use existing publicly available datasets such as the iNaturalist 2021 Mini Dataset or Animal Faces dataset (AFHQ), which contains thousands of labeled images across various species. We will use a filtered subset of these datasets (e.g., 10-20 classes) to keep training manageable given our computing resources.

Model Architecture and Code:

We will use a pre-trained deep convolutional neural network such as ResNet-50, Single Shot MultiBox Detector (SSD) or EfficientNet, available through the PyTorch torchvision.models module. These models are already trained on large image datasets like ImageNet and will be fine-tuned on our animal dataset using transfer learning.

Solution Design:

We will preprocess the data by resizing images and applying augmentations such as rotation, flipping, and color jitter. Using PyTorch, we will load a pre-trained CNN and replace the final classification layer to match our selected animal classes. The model will be fine-tuned on our dataset, and we will evaluate it using accuracy, precision, recall, and confusion matrices on a held-out test set. Or, we will load a pre-trained SSD model (e.g., SSD300 with VGG16 backbone) and fine-tune it on a selected subset of the Caltech Camera Traps dataset. The model will be trained to detect and classify animals by predicting bounding boxes and associated class labels. We will evaluate performance using standard object detection metrics such as mean Average Precision (mAP) and Intersection over Union (IoU) on a held-out test set.

Training Resources:

Because it is a small scale project, we will use Google Colab with GPU acceleration to train and evaluate our model.

Expected Outcome:

A good outcome would be achieving at least 85% classification accuracy on the test set.