Assignment_3 – Object Detection

Due – Dec 6, 2024

Submission instructions in Lecture_1

Context:

This project uses a subset of the Caltech101 dataset to train and evaluate an object detection model. The task involves detecting objects and predicting their bounding boxes and class labels. The dataset consists of three classes: Butterfly (91 images), Dalmatian (67 images), and Dolphin (65 images). This assignment provides an opportunity to explore object detection and classification in real-world contexts like wildlife conservation.

Part 1 - Assignment Instructions- Building a Python notebook – 100%

1. Importing Necessary Libraries
    - Load libraries such as TensorFlow/Keras, NumPy, Pandas, and OpenCV for deep learning and data manipulation.
2. Reading and Preparing the Dataset
    - Unzip the dataset.
    - Extract filenames and bounding box annotations for the three classes: Butterfly, Dalmatian, and Dolphin.
3. Scaling Bounding Box Coordinates
    - Scale bounding box coordinates relative to the input image dimensions.
4. Visualizing Sample Images
    - Display a subset of images from each class with their bounding boxes drawn to understand the dataset.
5. Data Preprocessing
    - Convert images from the range [0, 255] to [0, 1] and scale bounding box annotations accordingly.
    - Convert class labels into one-hot encoded vectors.
6. Model Preparation
    - Use pre-trained models such as VGG16, VGG19, or ResNet with the following modifications:

        - Scale inputs to match the model's requirements.
        - Construct fully connected layers for:
            - Predicting bounding box coordinates.
            - Predicting class labels.

7. Defining Loss Functions
    - Include separate loss functions for bounding box prediction and class label prediction.
    - Combine these losses during training.
8. Model Summary
    - Print the model summary, including the number of trainable parameters.
9. Training the Model
    - Fit the model on the training set, tracking the bounding box loss, classification loss, and respective accuracies across epochs.
10. Model Evaluation
    - Test the model on unseen images and visualize:
        - Bounding box predictions.
        - Class label predictions.
    - Display losses and accuracies per epoch.
11. Saving and Loading the Model
    - Save the trained model to a .pkl file.
    - Load the model from the saved file and evaluate it on three test images from each class.