Sunday, 24 November 2024

Machine Learning Engineering Assignment

Kuhu Sharma

Se21ucse109

Task 1: Dataset Preparation, Augmentation, and Hash Calculation

1. Introduction

Used a vehicle detection dataset from Kaggle and prepare it for training a YOLOv4 model. The task is broken down into three main steps:

1. **Dataset Download and Preprocessing**: Download a vehicle detection dataset, organize it into a proper format, and split it into training, validation, and test sets.
2. **Data Augmentation**: Implement data augmentation techniques to augment the dataset and visualize the augmented images.
3. **Hash Calculation**: Calculate the hash value of my name using a given Python code.

2. Dataset Preparation

2.1 Dataset Selection

To get the object detection dataset, we used the **Vehicle Detection Dataset** from Kaggle. You can access the dataset here: Vehicle Detection Dataset on Kaggle

2.2 Data Organization

Once the dataset was downloaded, it consisted of images and their corresponding annotation files. The images were categorized by the type of vehicle (e.g., car, bus, motorcycle), and the annotations were in a specific format (YOLO format).

To prepare the dataset for training with YOLOv4, I wrote a Python script to rearrange the images and annotations into the required folder structure.

2.3 Data Split

We selected **5 categories** from the dataset and divided them into **100 images per category**. After selecting the images, the dataset was split into the following ratio:

- **70% for training**
- **20% for validation**
- **10% for testing**

A Python script was used to randomly shuffle the images and annotations, and then assign them to the respective directories for training, validation, and testing.

3. Data Augmentation

3.1 Augmentation Techniques

To enhance the diversity of the dataset and improve the model's ability to generalize, we applied several data augmentation techniques. We used the **`albumentations`** library to implement these augmentations.

The following augmentation techniques were applied:

- **Random Crop**: Crops a random section of the image.
- **Horizontal Flip**: Flips the image horizontally with a probability of 50%.
- **Brightness and Contrast Adjustment**: Randomly adjusts the brightness and contrast of the image.
- **Resize**: Resizes the image to a fixed size (416x416 pixels) suitable for YOLOv4 input.
- **Random Sized Bounding Box Crop**: Randomly crops a bounding box from the image while ensuring the object remains visible.

4. Hash Calculation

4.1 Purpose

In this step, we were tasked with calculating the hash value of my name using Python's `hashlib` library. This ensures a unique numeric representation of the string input.

5. Conclusion

5.1 Summary

In this assignment, I successfully:

- **Downloaded and organized the dataset** into the required folder structure.
- **Applied basic data augmentation techniques** such as cropping, flipping, and adjusting brightness and contrast.
- **Calculated the hash value** of my name using Python's `hashlib` library.

5.2 Challenges Encountered

I encountered several challenges, particularly in the following areas:

- Understanding the YOLOv4 model setup and repository, which took longer than expected.
- Implementing data augmentation correctly and handling bounding boxes with transformations.
- Time constraints and the complexity of setting up an end-to-end object detection pipeline.

5.3 Willingness to Learn

Although I couldn't fully implement the object detection training and testing pipeline in this assignment, I am committed to learning and improving. I plan to:

- Dive deeper into object detection frameworks like YOLOv4 and its implementation.
- Gain more hands-on experience with data augmentation and its impact on model performance.
- Seek out additional resources and tutorials to strengthen my understanding of computer vision techniques.