**Animal Detection Using Faster R-CNN**

**Problem Definition**

Object detection is a critical task in computer vision that involves identifying and localizing objects in images. This project focuses on detecting various animal species (e.g., birds, cats, deer) using a Faster R-CNN model. For instance, in an image containing a bird and a cat, the model aims to draw bounding boxes around each animal and correctly label them. The importance of Animal detection spans multiple fields, such as wildlife monitoring, farming, detecting animals near crops can prevent damage and reduce conflicts between humans and wildlife

**Dataset Description**

The dataset used in this project comprises images of animals organized into three sets: training, validation, and testing. Each image is annotated with bounding boxes and corresponding class labels. The data was organized into subfolders based on class names, and the annotations were stored in CSV files with normalized coordinates for bounding boxes. Key statistics of the dataset are as follows:

| **Dataset Split** | **Number of Images** |
| --- | --- |
| Training | 3,000 |
| Validation | 1500 |
| Testing | 1000 |

**Class Distribution**

The dataset contains 10 classes: Bird, Cat, Deer, Dog, Fish, Giraffe, Lion, Owl, Sheep, and Tiger. The classes are balanced to ensure equal representation across splits.

**Deep Learning Approach**

This project employs the Faster R-CNN model with a ResNet-50 backbone and Feature Pyramid Network (FPN). The model was pre-trained on the COCO dataset to leverage transfer learning and fine-tuned on the custom dataset. The choice of Faster R-CNN was motivated by its state-of-the-art performance in object detection tasks, offering a good trade-off between speed and accuracy.

Key modifications:

* **Classifier Head:** The classifier was modified to output predictions for 11 classes (10 animal classes + background).
* **Input Image Resizing:** Images were resized to a fixed size of 512x512 pixels to ensure consistency and reduce computational load.
* **Optimizer:** AdamW optimizer was used with a learning rate of 0.0001 and weight decay of 0.0001 to enhance convergence.

**Evaluation Metrics**

The following metrics were used to evaluate the model's performance:

1. **Mean Average Precision (mAP):** Measures detection accuracy across all classes.
2. **Precision:** Indicates the percentage of correct predictions out of all predictions.
3. **Recall:** Reflects the percentage of true positives identified out of all actual positives.

**Hyperparameter Tuning Process**

Hyperparameter tuning was conducted systematically:

* **Learning Rate:** Tested values were [0.0005, 0.001, 0.0001], with 0.0001 yielding the best results.
* **Batch Size:** A batch size of 16 was chosen to balance memory constraints and training efficiency.
* **Optimizer:** SGD and AdamW were tested, with AdamW providing better convergence and stability.

Insights:

* Lower learning rates ensured stable training without overshooting minima.
* AdamW’s weight decay improved generalization by preventing overfitting.

**Results and Discussion**

**Quantitative Results**

The model achieved the following performance metrics on the test set:

**Qualitative Analysis**

The model successfully detected and labeled multiple objects in complex scenes. However, challenges included:

* Overlapping objects leading to missed detections.
* False positives for similar-looking classes (Lion vs. Tiger).

|  |  |
| --- | --- |
| metric | Value |
| mAP | 0.7419 |
| Recall | 0.9979 |
| precision | 0.7419 |

**Challenges Encountered**

1. **High False Positive Rate:** This was mitigated by increasing the IoU threshold during evaluation.
2. **Compute Constraints:** Training the model on large datasets required significant computational resources. Using a GPU accelerated the process, but further optimization was necessary to reduce training time.

**Visualizations**

Sample predictions with bounding boxes and class labels highlighted the model’s ability to accurately localize and identify objects. Bounding boxes were adjusted to resized dimensions (512x512) for consistency.

**Conclusion**

The Faster R-CNN model demonstrated strong performance in detecting and classifying animals in images. The use of transfer learning, coupled with careful dataset preprocessing and hyperparameter tuning, was critical to achieving high accuracy. While the model performed well overall, future work could address challenges such as overlapping objects and false positives by exploring advanced techniques like Soft-NMS or multi-scale training. Additionally, augmenting the dataset with more diverse examples could further enhance generalization.