Assignment 1 - Simplified Custom Object Detection

Submit Assignment

Due Apr 21 by 11:59pm **Points** 5
Submitting a text entry box, a website url, or a file upload
Available Mar 13 at 1pm - Apr 21 at 11:59pm about 1 month

Assignment 1 - Simplified Custom Object Detection

This assignment is designed for the Deep Network Development course and consists the creation and evaluation of a custom object detection model alongside the comparison with an existing fine-tuned object detection model.

General Instructions:

- Template/Guideline notebook: https://colab.research.google.com/drive/1epFtlmlkTYZAAKhiuYgiHQgiCnL3wDD4?usp=sharing (https://colab.research.google.com/drive/1epFtlmlkTYZAAKhiuYgiHQgiCnL3wDD4?usp=sharing)
- **Submission Deadline: April 21st, Sunday, 11:59 PM.** Late submissions will not be accepted.
- Submission Method: Upload the completed .ipynb file to Canvas.
- Policy: Copying code from others will result in an automatic failure of the assignment
- **Assignment Defense: To receive a grade for the assignment, you must be prepared to answer questions during the designated practice time.**

Task Description:

Your task is to develop and train a custom object detection model and then compare its performance with an existing object detection model (i.e. YOLO). This assignment focuses on a simplified version of object detection, where each image contains only a single object. The dataset used for training will be synthetically generated. Here are the specific tasks:

1. **Dataset Creation:**

- Select at least 3 statues from Mihály Kolodko's Mini Statues (link in template notebook).
- Remove the background from the chosen objects, ensuring only the object remains.
- Collect background images from different locations in Budapest.
- Generate a synthetic dataset by randomly inserting one object into a background image at a random location.
- Save the coordinates of the inserted objects to create bounding boxes.
- Format the bounding boxes appropriately for training (e.g., [x,y,w,h], or YOLO format).
- Implement a PyTorch Dataset class to manage the dataset.
- Split the dataset into training, validation, and test sets.

2. **Model Development:**

- Design a Convolutional Neural Network (CNN) based architecture for object detection.
- Include a backbone for feature extraction and two output branches:

    - One for class probabilities (equal to the number of chosen objects).
    - Another for bounding box regression (of size 4 for coordinates).
- Experiment with different layers and hyperparameters, incorporating regularization techniques.
- Define an optimizer, loss function, and number of epochs for training.
- Optimize both classification and bounding box regression losses.
- Choose at least 3 evaluation metrics for your model (e.g., Precision, Recall, F-1, mAP).
- Visualize performance metrics and model predictions.

3. **Existing Model Comparison:**

- Load an existing object detection model (e.g., YOLOv8).
- Fine-tune the model on the synthetic dataset created earlier.
- Monitor and visualize losses and metrics during fine-tuning.
- Compare the performance of the custom model and the fine-tuned existing model.
- Visualize predictions from both models and justify any observed differences.
- Provide insights on potential improvements for the custom model.

Guidelines:

- Follow the sections provided in the notebook to guide your implementation, but feel free to adapt as needed.
- Carefully read each instruction and ensure all requirements are met.
- Avoid using external Python files for code. All code should be contained within the notebook.
- Include your name, Neptun ID, network details, and chosen objects at the beginning of the notebook.
- Add explanations throughout your code and in the results section, detailing your choices, observations, and potential improvements.

Defense Preparation (some example questions, there are more):

- Explain what is object detection.
- What are the key components of a convolutional neural network (CNN) architecture commonly used in object detection?
- How does the process of data augmentation contribute to improving the performance of an object detection model?
- Describe the role of anchor boxes in object detection algorithms such as YOLO and SSD.
- Can you explain the difference between localization and classification in the context of object detection?
- How do you evaluate the performance of an object detection model, and what are some commonly used metrics?
- What are the main challenges faced when training a custom object detection model, and how can these challenges be addressed?
- Explain the concept of transfer learning and how it can be applied in the context of fine-tuning pre-trained object detection models.
- What are some techniques used to prevent overfitting in object detection models, and why are they important?
- Can you discuss the trade-offs between speed and accuracy when choosing an object detection model architecture, especially in real-time applications?

By following all these instructions and delivering a well-structured and documented notebook, you'll demonstrate your understanding of object detection techniques and your ability to implement and evaluate custom models effectively. Good luck with your assignment!