Implementation Report

Table of Contents

Introduction

Object detection is a pivotal task in computer vision, enabling machines to identify and locate objects within images. This report details the implementation of an object detection system tailored for indoor environments, specifically focusing on detecting tables, trash cans, and chairs. Leveraging the state-of-the-art YOLOv7 (You Only Look Once version 7) architecture, the project encompasses dataset preparation, model training, evaluation, and visualization of results. Additionally, the system incorporates distance estimation based on bounding box dimensions to provide spatial context for detected objects.

Dataset Description

Dataset Source

The dataset utilized for this project comprises annotated images focusing on indoor environments. The dataset was sourced from a publicly available repository and further annotated to suit the specific classes of interest: tables, trash cans, and chairs.

Annotation Format

Annotations are provided in a standardized format where each image has an accompanying text file containing bounding box coordinates and class labels. The format follows the YOLO annotation style, which includes:

- **Class ID**: Numerical identifier for each object class.
- **Bounding Box Coordinates**: Normalized values representing the center (x, y), width, and height of the bounding box relative to the image dimensions.

This structure facilitates seamless integration with YOLOv7, enabling efficient training and inference processes.

Class Labels

The dataset encompasses three distinct classes:

1. **Table**
2. **Trash Can**
3. **Chair**

These classes were selected to represent common indoor objects, providing a focused scope for the detection system.

Data Partitioning

To ensure robust training and evaluation, the dataset was partitioned into three subsets:

- **Training Set**: 80% of the data, used for model training.
- **Validation Set**: 10% of the data, employed for tuning hyperparameters and preventing overfitting.
- **Test Set**: 10% of the data, reserved for final model evaluation and performance assessment.

This partitioning strategy ensures that the model generalizes well to unseen data and maintains high performance across different subsets.

Data Preprocessing

Preprocessing steps were meticulously applied to prepare the data for training:

- **Normalization**: Image pixel values were normalized to a standard range to ensure consistent input for the model.
- **Transformation**: Data augmentation techniques such as scaling, rotation, and flipping were employed to enhance the model's robustness and ability to generalize across various scenarios.

These preprocessing techniques are crucial for improving model performance and ensuring resilience against variations in real-world data.

Methodology

Model Selection

The YOLOv7 architecture was selected for this project due to its state-of-the-art performance in object detection tasks. YOLOv7 is renowned for its balance between speed and accuracy, making it suitable for real-time applications. Its single-stage detection mechanism allows for efficient processing, while its architectural advancements contribute to superior detection capabilities.

Environment Setup

The implementation environment was configured using a combination of system packages and Python dependencies:

- **System Packages**: Essential tools such as wget, git, and unzip were installed to facilitate repository cloning and data handling.
- **Python Dependencies**: Critical libraries and frameworks required by YOLOv7 were installed to ensure seamless integration and functionality.

This comprehensive environment setup was fundamental in establishing a stable and efficient workflow for model training and evaluation.

Model Configuration

To tailor YOLOv7 to the specific detection task, the model configuration was adjusted:

- **Number of Classes (nc)**: The configuration file was modified to reflect the three classes of interest, replacing the default value (typically 80 for COCO dataset) with nc: 3.
- **Dataset Paths**: Paths to the training, validation, and test datasets were specified in a custom YAML configuration file (chairs.yaml), ensuring the model correctly accesses and utilizes the dataset.

These configurations are pivotal in aligning the model architecture with the dataset's characteristics, enabling accurate training and detection.

Training Process

The training pipeline encompassed the following steps:

1. **Initialization**: Pre-trained YOLOv7 weights were utilized as a starting point to leverage learned features and accelerate convergence.
2. **Training Execution**: The model was trained over 50 epochs with a batch size of 8, balancing computational efficiency and learning stability.
3. **Monitoring**: Training progress was tracked, focusing on loss metrics to assess learning and adjust training parameters as necessary.

This structured training approach facilitated the development of a robust detection model capable of accurately identifying and localizing objects within indoor environments.

Loss Function

YOLOv7 employs a composite loss function that integrates several components to optimize detection performance:

- **Localization Loss**: Measures the accuracy of bounding box predictions, ensuring precise localization of objects.
- **Confidence Loss**: Evaluates the model's confidence in the presence of objects within predicted bounding boxes.
- **Classification Loss**: Assesses the accuracy of class predictions for detected objects.

This multifaceted loss function ensures comprehensive optimization, balancing the trade-offs between localization precision, confidence calibration, and classification accuracy.

Evaluation

Evaluation Metrics

The model's performance was evaluated using standard object detection metrics, including:

- **Mean Average Precision (mAP)**: Measures the average precision across all classes, providing a holistic assessment of detection accuracy.
- **Precision and Recall**: Evaluate the model's ability to correctly identify positive instances and retrieve relevant detections, respectively.
- **Loss Metrics**: Track the training and validation loss to monitor learning progress and detect overfitting or underfitting.

These metrics offer a comprehensive understanding of the model's detection capabilities and overall performance.

Validation and Testing

Post-training, the model was evaluated on both validation and test datasets:

- **Validation Set**: Utilized for hyperparameter tuning and early stopping to prevent overfitting.

- **Test Set**: Employed for final performance assessment, ensuring the model's generalizability to unseen data.

This dual-phase evaluation approach ensures that the model not only performs well on known data but also maintains high accuracy on novel inputs.

Results

The training process yielded the following outcomes:

- **Average Training Loss**: The mean training loss across epochs indicated consistent learning and optimization.
- **Average Validation Loss**: The mean validation loss closely followed the training loss, suggesting good generalization and minimal overfitting.

These results reflect the model's effective learning and its ability to generalize from training data to unseen validation and test sets.

Distance Estimation

An additional feature implemented was distance estimation based on bounding box dimensions:

- **Calculation Method**: Distance was estimated using the width and height of the bounding boxes, applying a mathematical formula to derive spatial information.
- **Purpose**: Provides contextual understanding of object placement and spatial relationships within the indoor environment.

This functionality enhances the detection system by adding a layer of spatial analysis, beneficial for applications requiring depth perception and object localization.

Visualization of Results

Visual representations of detection results were generated to illustrate the model's performance:

- **Bounding Boxes**: Detected objects were highlighted with bounding boxes, annotated with class labels and estimated distances.
- **Sample Displays**: A selection of images from the test set showcased the model's ability to accurately identify and localize objects, alongside distance estimations.

These visualizations serve as intuitive validations of the model's detection accuracy and spatial analysis capabilities.

Conclusion

This implementation successfully developed an object detection system tailored for indoor environments, leveraging the YOLOv7 architecture. The project encompassed comprehensive dataset preparation, model configuration, training, and evaluation, culminating in a robust detection model capable of identifying tables, trash cans, and chairs with high accuracy. Additionally, the incorporation of distance estimation augments the system's utility by providing spatial context for detected objects.

Future work may explore expanding the class repertoire, enhancing distance estimation algorithms, and optimizing model performance for real-time applications.

References

- Jocher, G., et al. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *GitHub Repository*. Retrieved from https://github.com/WongKinYiu/yolov7
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*.
- Russakovsky, O., et al. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 211-252.

SharePoint link

https://unhnewhaven-my.sharepoint.com/:f:/g/personal/rmanu4_unh_newhaven_edu/EnsoAb2Gf8VMmmHPDkFF9ngBAer9ukusaC4O3wEfj4_mcQ?e=sejlyf