Object Detection Cheat Sheet

Devin Prikryl-Martin
Shakira Kashif
Christian Mpabuka
Piero Suarez-Prada
Houston Community College
ITAI 1378 Computer Vision
Professor Patricia McManus
October 24, 2024

Steps involved in a typical object detection task:

1.Input- The model receives input in the form of an image or video and preprocesses it (resizing, normalizing, etc) so that it is easier to analyze. ***2.Feature extraction-*** The model uses a combination of techniques to look for patterns (features) in the image, such as finding edges, corners, textures, shapes and colors. ***3.Region proposal-*** The model predicts where objects are in an image and draws bounding boxes around predicted objects. ***4.Classification and localization-*** For each predicted object region the model classifies the object and supplies a confidence score for how sure it is about the accuracy of the prediction. The box is sometimes adjusted to better fit the object. ***5.Post-processing-*** The model reviews its predictions and removes duplicate detections, predictions it is less confident in, and unlikely predictions.***6.Output-*** The model outputs the image with boxes around the detected objects. These boxes are attached to identifying labels and corresponding confidence scores.

Common challenges and how to overcome them:

Data issues: *Limited data, unbalanced datasets where some classes have many more samples, and poorly/inconsistently labeled data. Solutions: using data augmentation or synthetic data creation to increase quality training data, using pre-trained models, and using semi-supervised self-learning approaches to improve annotations.****Performance issues****: high computational requirements, memory constraints, poor detection of small objects and problems with overlapping bounding boxes. Solutions: optimizing (pruning, quantizing etc) your model so it requires less computation, using a lighter architecture like YOLO or SSD, increasing your input resolution and using feature pyramid networks so small objects are easier to detect.*

Definitions:

Bounding Boxes: *In AI/computer vision, these are rectangular boxes defined by (x, y) coordinates that outline the location of detected objects in an image. They're typically represented by four values: x and y coordinates of the top-left corner, width, and height.* ***Annotations:*** *The ground truth or human-labeled data that marks the correct locations and classifications of objects in training images. For object detection, this typically includes both bounding box coordinates and class labels. These serve as the "correct answers" during model training.* ***Intersection over Union (IoU):*** *A metric that measures the accuracy of predicted bounding boxes by comparing them to ground truth boxes. It's calculated by dividing the area of overlap (intersection) by the area of union between the predicted and ground truth boxes. Values range from 0 to 1, with 1 being a perfect overlap.* ***Confidence Scores:*** *Probability values (0-1) assigned by the model to each detection, indicating how certain it is about both the object's presence and its classification. Higher scores indicate greater confidence in the prediction.* ***R-CNN (Region-based Convolutional Neural Network):*** *The original region-based CNN architecture that uses selective search to propose regions, then applies a CNN to classify each region. While groundbreaking, it's computationally expensive as it processes each region separately.* ***Fast R-CNN:*** *An improvement over R-CNN that processes the entire image through a CNN first, then extracts features for each region proposal from the shared feature map. This significantly speeds up training and inference compared to R-CNN.****Faster R-CNN:*** *Further improves Fast R-CNN by replacing selective search with a Region Proposal Network (RPN), making the entire system end-to-end trainable. This network learns to generate region proposals directly from the feature maps.* ***SSD (Single Shot Detector):*** *A one-stage detector that eliminates the need for region proposals by predicting bounding boxes and class probabilities directly from feature maps at multiple scales. This makes it faster than two-stage detectors like Faster R-CNN.* ***YOLO (You Only Look Once):*** *Another one-stage detector that divides the image into a grid and predicts bounding boxes and class probabilities directly. Known for its speed and real-time performance, though sometimes with lower accuracy than two-stage detectors. Has evolved through multiple versions (v1-v8) with significant improvement.*

Tools and libraries:

TensorFlow*: is an open-source machine learning library developed by Google that is extensively used for building, training, and deploying deep learning models. It supports a wide range of machine learning tasks, from image classification and object detection to natural language processing.****Applications: Real-time Object Detection:*** *For tasks like pedestrian detection in autonomous vehicles or facial recognition systems.* ***Installation****: pip install tensorflow* ***link****: TensorFlow* ***Keras****: is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. Keras also gives the highest priority to crafting great documentation and developer guides.* ***Example Applications: Custom Object Detectors:*** *You can create custom detectors for niche tasks, such as identifying specific animals in wildlife camera images or detecting defects in industrial components.* ***Installation:*** *pip install tensorflow* ***link****: Keras: Deep Learning for humans* ***OpenCV (Open-Source Computer Vision Library)****: is an open-source computer vision and machine learning software library. It provides real-time computer vision capabilities and is commonly used for tasks like image and video processing, face detection, and object tracking.* ***Example Applications: Real-Time Object Detection in Video Feeds:*** *Common security systems for recognizing intruders or identifying specific objects.* ***Installation:*** *pip install opencv-python* ***Link:*** *OpenCV - Open Computer Vision Library* ***Darknet****: is an open-source neural network framework written in C and CUDA. It's the framework behind the YOLO (You Only Look Once) family of models. Darknet is designed for performance and ease of use, making it a preferred choice for real-time object detection. Optimized for performance on both CPUs and GPUs (CUDA acceleration).Pre-Trained Models: YOLOv4 and YOLOv5 are commonly used and provide a good balance of speed and accuracy.* ***Example Applications: Autonomous Vehicles:*** *YOLO is used for detecting pedestrians, vehicles, traffic signs, and other objects in the driving environment in real-time.* ***Link****: Darknet: Open Source Neural Networks in C*

Additional Resources:

1. Image Processing, Analysis and Machine Vision by Milan Sonka, Vaclav Hlavac and Roger Boyle.
2. First Principles of Computer Vision by Shree Nayar.
3. Dive into Deep Learning by Aston Zhang, Zachary C. Lipton, Mu Li and Alexander J. Smola.

Brief reflection:

The field of object detection encompasses a large amount of information such as the steps that the detection process takes, the different problems that can occur during the performance of a detection task and the possible tools that can be used for the various applications of object detection. It is common that during learning we cannot retain in our mind all the information we receive. Rapid access to already processed information can facilitate the retention of information during the learning process of knowledge already learned or yet to be learned. Creating a cheat sheet can be helpful in finding keywords that we might forget and need to remember as quickly as possible. In this cheat sheet we also added some additional resources such as books and courses that, if necessary, can be an extra source of information to expand our knowledge without the need to increase the amount of information contained in the sheet.