**Definition**

**Object Detection Algorithms**

**Tools and Library**

|  |  |
| --- | --- |
| Bounding Boxes | + define location of objects within an image  + (x, y, width, height)  + localize objects  Goals: Predict coordinates to place boxes around detected objects |
| Annotations | + process of labeling image to provide info about the objects in the image |
| Confidence Score | +Score between 0 and 1 to indicate how confident the model is |
| Classes | +Categories of objects that the model is trained to detect. Each object is assigned a class label |
|  |  |

|  |  |  |
| --- | --- | --- |
| PyTorch | Emphasizes flexibility and ease of use, popular in research, and supports various object detection models. |  |
| TensorFlow | Developed by Google, it offers robust support for building and training neural networks, supports various object detection models (e.g., TensorFlow Object Detection API), and includes pre-trained models. |  |
| Keras | A high-level neural networks API built on TensorFlow, simplifying deep learning model development. Commonly used for prototyping and integrates with TensorFlow for training object detection models. |  |
| OpenCV | Useful for preprocessing images and integrating with deep learning models for inference. |  |

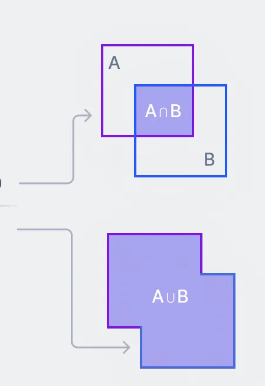
|  |  |
| --- | --- |
| R-CNN | +uses selective search to propose candidate object regions and applies a CNN to classify each region. |
| Fast R-CNN | +An improvement over R-CNN, Fast R-CNN processes the entire image through a CNN first, then classifies proposals from a feature map. |
| Faster R-CNN | +Region Proposal Network (RPN) to generate region proposals within the CNN itself, significantly speeding up the process. |
| SSD | +A single-shot detection method that predicts bounding boxes and class scores directly from feature maps at different scales |
| YOLO | + A real-time object detection system that frames the problem as a regression task, predicting bounding boxes and class probabilities directly from full images in one evaluation. |

**Installation Instruction**

|  |  |
| --- | --- |
| PyTorch | +pip install torch torchvision torchaudio  <https://pytorch.org/> |
| TensorFlow | +pip install tensorflow  <https://www.tensorflow.org/> |
| Keras | +pip install keras  <https://keras.io/> |
| OpenCV | +pip install opencv-python  <https://opencv.org/> |

**Intersection over Union (IoU)**

|  |  |
| --- | --- |
| Definition | +a key metric in computer vision for evaluating object detection accuracy. It measures the overlap between the "ground truth" bounding box and the model's predicted box, indicating how well the prediction aligns with the actual object. A higher IoU score means better accuracy. |
| Calculation |  |
| Common threshold | +IoU >= 0.5 -> predicted box overlaps with the ground truth at least 50% |



A butterfly with purple squares

Description automatically generated

**Difference of IoU Threshold**

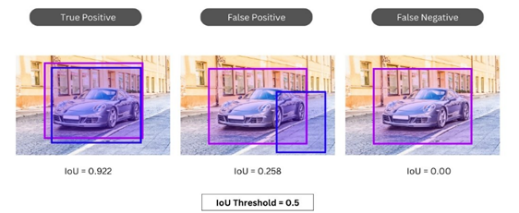
**Challenges Troubleshoots**

|  |  |
| --- | --- |
| True Positive (TP) | +a predicted bounding box with a high enough IoU (0.5+) |
| False Positive (FP) | +a predicted bounding box that does not overlap significantly with any ground truth box |
| False Negative (FN) | +a ground truth box that the model missed the entirely, fail to detect an existing object |

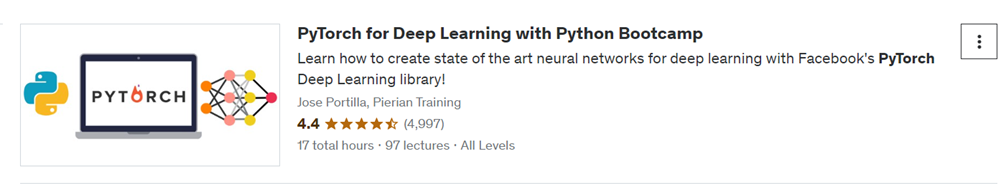
|  |  |
| --- | --- |
| + Objects can be large, small, or oddly shaped | +Use Faster R-CNN/SSD to handle varying object sizes  +Apply image pyramids to detect objects at different scales  +Use anchor boxes with different aspect ratios and sizes to improve detection |
| +Objects might be partially hidden by other objects. | +Implement context-aware models to consider the surrounding area of object  +Data augmentation techniques like random cropping to train the model for occlusions  +Apply NMS to prevent overlapping bounding boxes from being suppressed incorrectly |
| +Different lighting conditions or complex backgrounds can make detection difficult. | +Data augmentation to simulate various lighting conditions during training  +Apply preprocessing to normalize lighting across  +Use robust model like YOLO to perform better  +Use background subtraction to reduce the impact |

**Object detection step-by-step tasks**

|  |  |
| --- | --- |
| Data Collection | Gather varied images with target objects |
| Annotation | Images with bounding boxes & Class names |
| Preprocessing | Resize and Normalize images to establish a consistency |
| Model Selection | Choose a suitable detection model |
| Training | Train model on the annotated dataset |
| Evaluation | Measure performance using metrics |
| Fine-tuning | Adjust the model for improved accuracy |
| Deployment | Implement the model in real-world apps |
| Monitoring | Track performance and update |



**Additional Resources**



Group 1: Andre Ellis Carlos Granillo

Khanh Huynh

Oluwatoyin Alobarin Sharise Griggs Enrique Quintero Garcia