**A09 ITAI 1378 CV**

**Cheat Sheet**

**Team EyeConic**

**Brooke Broderick - W217065417**

**Matthew Choo - W206744145**

**Cameroun White-W216530571**

**Melvis Maduagwu-W216890463**

**Erick Banegas - W2162009**

**NancyChieu-W000098778**

**Yoana Cook - W215913890**

**6252-ITAI-1378-Comp Vision-Artificial Intel-RT-15698 - Spring 2025**

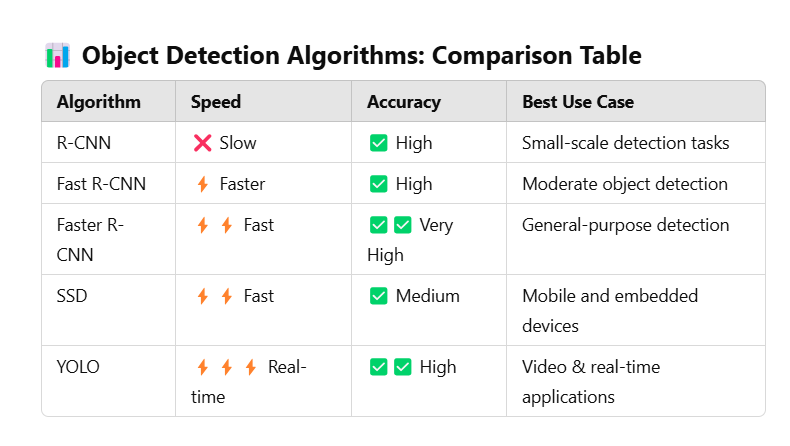
**Professor Anna Devarakonda**

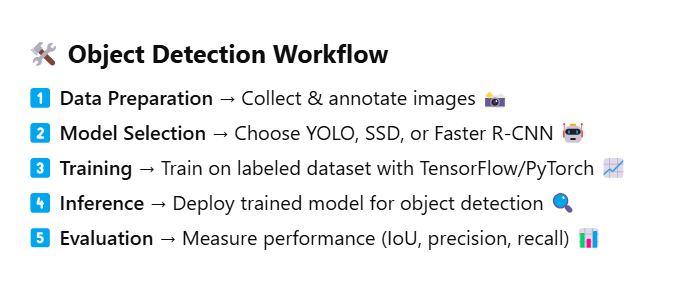
**March 22, 2025**

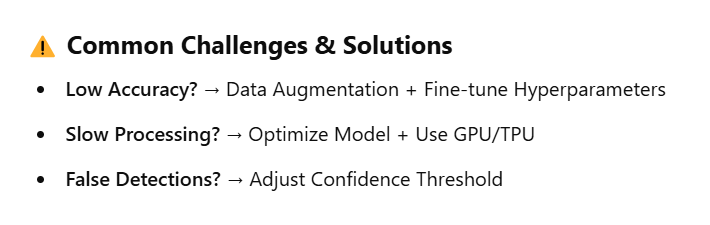
| Yoana’s cheat sheet: | |
| --- | --- |
| **Key Concepts in Object Detection:** | |
| **Annotations:** | Labels or metadata added to images to indicate object locations and classes. |
| **Bounding Boxes:** | Rectangular frames used to locate and identify objects within an image. |
| **Confidence Scores:** | A measure of how confident the model is in its detection (e.g., 0.95 means 95% confidence). |
| **Intersection over Union (IoU):** | A metric to evaluate detection accuracy by measuring the overlap between predicted and ground-truth bounding boxes. |

| **Common Object Detection Algorithms:** | |
| --- | --- |
| **R-CNN (Region-based Convolutional Neural Network):** | Detects objects by proposing regions and classifying them. |
| **Fast R-CNN:** | Improves R-CNN by processing the entire image with a CNN before proposing regions. |
| **Faster R-CNN:** | Enhances speed by using a Region Proposal Network (RPN) to generate region proposals. |
| **SSD (Single Shot Detector):** | Detects objects in a single pass through the network, balancing speed and accuracy. |
| **YOLO (You Only Look Once):** | A real-time object detection algorithm that processes images in one forward pass. |

| **Tools and Libraries:** | |
| --- | --- |
| **TensorFlow/Keras:** | Popular frameworks for building and training deep learning models. |
| **OpenCV:** | A library for image processing and computer vision tasks. |

****

****

****

**Suggested Diagrams to Include**

**Bounding Box Example**

**Show an image with objects enclosed in labeled bounding boxes.**

**Example: A car labeled with Car (90%), a person labeled Person (85%).**

**Intersection over Union (IoU) Calculation**

**Draw two overlapping rectangles: one representing the ground truth and the other the predicted bounding box.**

**Label the overlapping area and the union area to illustrate the IoU formula.**

**Comparison of Object Detection Algorithms**

**Create a flowchart showing how R-CNN, Fast R-CNN, Faster R-CNN, SSD, and YOLO process images differently.**

**Example:**

**R-CNN → Extract regions → CNN on each region → Classification**

**YOLO → Single neural network pass → Predicts bounding boxes & classes.**

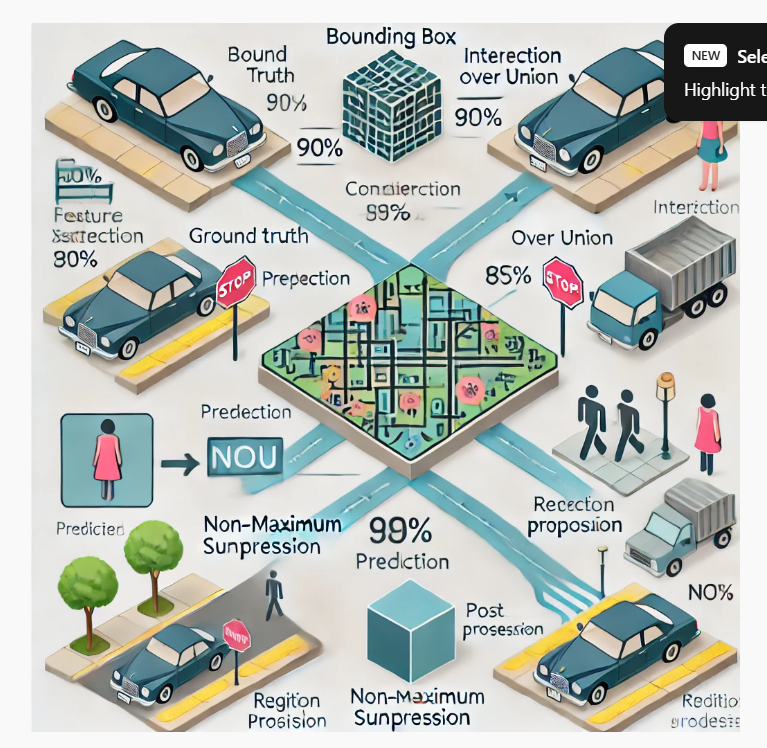
**Object Detection Pipeline**

**Step-by-step flowchart with labeled boxes:**

**Input Image → Preprocessing → Feature Extraction → Region Proposal → Classification → Post-Processing.**

**Non-Maximum Suppression (NMS)**

**Show multiple overlapping bounding boxes and how NMS selects the best one based on confidence scores.**

****

**Here’s a diagram illustrating key object detection concepts, including bounding boxes, IoU calculation, the object detection pipeline, and Non-Maximum Suppression (NMS).**

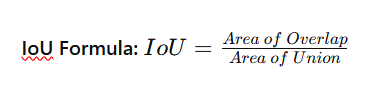
**Bounding Box: A rectangular box that defines the location of an object in an image.**

**Annotations: Labeled data used to train object detection models.**

**Confidence Score: A value indicating the model’s certainty in detecting an object.**

**Intersection over Union (IoU): A metric for evaluating object detection accuracy by comparing the predicted and ground truth bounding boxes.**

**IoU Formula:**

****

**Tools & Libraries Overview**

**TensorFlow & Keras – Deep learning frameworks for model training.**

**OpenCV – Image processing and real-time detection.**

**PyTorch – Another deep learning framework popular for research.**

**pip install tensorflow opencv-python**

**Tools & Libraries**

**TensorFlow & Keras**

* **Installation: pip install tensorflow**

**OpenCV. \*\***

* **Installation: pip install opencv-python**

**Object Detection Pipeline**

**Preprocessing: Resize images, normalize pixel values, and annotate training data.**

**Feature Extraction: Use CNN-based architectures to extract image features.**

**Region Proposal: Identify potential object locations (used in R-CNN models).**

**Classification & Regression: Classify objects and refine bounding boxes.**

**Post-Processing: Apply Non-Maximum Suppression (NMS) to remove duplicate detections.**

### **Common Challenges & Troubleshooting**

| **Challenge** | **Solution** |
| --- | --- |
| **Low Accuracy** | **Use data augmentation, fine-tune the model, or adjust hyperparameters.** |
| **Slow Inference** | **Use a lightweight model like SSD or YOLO.** |
| **Overfitting** | **Increase dataset size or use regularization techniques.** |

**Additional Resources**

**Books:** "Deep Learning for Computer Vision" by Adrian Rosebrock.

**Online Courses:** Coursera’s "Introduction to Object Detection with TensorFlow".

DeepBean. 2023, March 11. “How Yolo Object Detection Works”. YouTube. <https://youtu.be/svn9-xV7wjk?si=hXYnGZbOKJKaq5nE>

Edje Electronics. Accessed 2025, March 22. "How to Train YOLO Object Detection Models in Google Colab". YouTube. <https://youtu.be/r0RspiLG260?si=NiZgTef1kjDqLoWN>

LearnOpenCV. Accessed 2025, March 22. "YOLO-NAS: Introducing One of The Most Efficient Object Detection Algorithms".

YouTube. <https://youtu.be/iZGxBs3H7Bs?si=HcG4eKNIDHL2c0eD>

**Docs:** TensorFlow Object Detection API

**Reflection**

**Brooke Broderick**

* We created a cheat sheet to summarize key concepts in object detection. It covers essential tools like TensorFlow, OpenCV, and PyTorch, along with popular algorithms such as CNN, Fast R-CNN, Faster R-CNN, SSD, and YOLO. We learned that each algorithm has unique advantages—Faster R-CNN improves efficiency with Region Proposal Networks (RPN), while YOLO is designed for real-time detection. A key concept in our cheat sheet is Intersection over Union (IoU), which measures how well a predicted bounding box aligns with the actual object. The cheat sheet also outlines the object detection workflow, from collecting and labeling data to training, evaluating, and deploying models. It addresses common challenges like class imbalance, occlusions, and real-time processing needs, providing solutions such as data augmentation, feature pyramid networks (FPN), and optimization techniques like pruning and quantization. To support our learning, we included additional resources on computer vision, object detection, and TensorFlow.

**Matthew Choo**

* After working on the cheat sheet, we were able to understand the tools, library overview, and common object detection algorithms such as -CNN, Fast R-CNN, Faster R-CNN, SSD, and YOLO. Also learn about a formula called Intersection over Union (IoU) that is used to compare the predicted and ground truth bounding boxes. Have learned about the object detection algorithm, workflow and the common challenges & solutions. Was able to find the additional sources related to computer vision, object detection and tensorflow. Overall this cheat sheet helps us learn about these concepts and the terms of it.

**Cameroun White**

* For this assignment, we researched key concepts and common object detection algorithms, including R-CNN, Fast R-CNN, Faster R-CNN, SSD, and YOLO. We explored tools and libraries such as TensorFlow/Keras and OpenCV. As part of the process, we created a cheat sheet, comparison tables, diagrams, and flowcharts to visualize core concepts. We also identified common challenges and troubleshooting methods. This assignment provided a strong foundation in object detection for computer vision, with additional resources included.

**Melvis Maduagwu**

* This assignment has introduced me to object detection, a major part of computer vision. I learned about key components like bounding boxes, object classification, localization and confidence scores. I also learned about the different libraries and tools involved like tensor flow and openCV. Out of all the models, we used the SSD because it was decently fast and accurate and suitable for our level of computational resources. It also works well on mobile devices. Object detection is an interesting topic and its use cases are in our everyday lives. Examples of these are surveillance, autonomous vehicles and industrial applications.

**Erick Banegas**

* This project helped me understand object detection better, especially in the field of computer vision. I learned about tools like TensorFlow, OpenCV, and Keras, and studied algorithms like CNN, Fast R-CNN, Faster R-CNN, SSD, and YOLO. Each one has its own advantages—like YOLO being great for real-time tasks, and Faster R-CNN being more accurate.

I also learned key concepts like bounding boxes and Intersection over Union (IoU), which helps measure how well a model detects an object. Creating the cheat sheet helped me organize all this information in a clear and simple way, and I know I’ll be able to use it later as a quick reference.

We also explored common challenges like class imbalance and real-time processing, and learned some solutions like data augmentation and optimization techniques. Overall, this project gave me a solid base and made me more interested in learning and practicing more in this area.

**NancyChieu**

* Wehad to search for the information assigned to understand the key concepts, the common object detection algorithms for R-CNN, Fast R-CNN, Faster R-CNN, SSD, and YOLO, plus explore tools and libraries in TensorFlow/Keras, and OpenCV. In the process we created cheat sheets for core concepts, tools, libraries, comparison tables for algorithms, diagrams for key concepts, flowchart for object detection, and found common challenges and how to troubleshoot. This assignment is a good way to learn the basics for object detection in computer vision. Additional resources are included. This will serve us well in the future study on object detection.

**Yoana Cook**

* This assignment was a great way to explore into the world of object detection and break down its core concepts into something more digestible. I started by researching the basics, things, such as bounding boxes, annotations, confidence scores, and Intersection over Union (IoU). I also explored popular algorithms like R-CNN, Fast R-CNN, Faster R-CNN, SSD, and YOLO, which gave me a better understanding of how object detection has evolved over time. On the tools side, I got familiar with TensorFlow, Keras, and OpenCV, which are essential for actually putting these algorithms into practice.
* Creating the cheat sheet was where everything came together. Using the Cornell note-taking method to organize the research, which made it easier to structure the cheat sheet. This method helped keep the cheat sheet clean, concise, and easy to follow.
* One of the biggest takeaways was learning how to simplify complex ideas without losing their essence. The cheat sheet is not just a study tool, it is something I can refer back to in future projects to quickly recall important details or troubleshoot issues. Plus, the process of researching and organizing the information helped solidify my understanding of object detection as a whole. I am looking forward to using this knowledge in real-world tasks and exploring more advanced topics down the line.