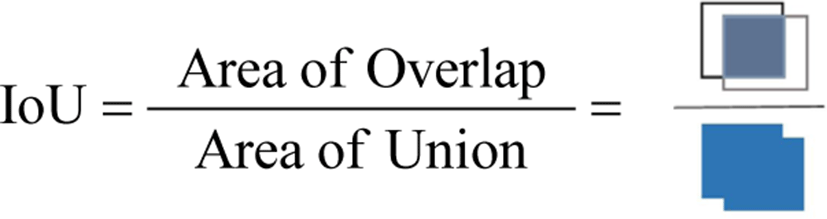
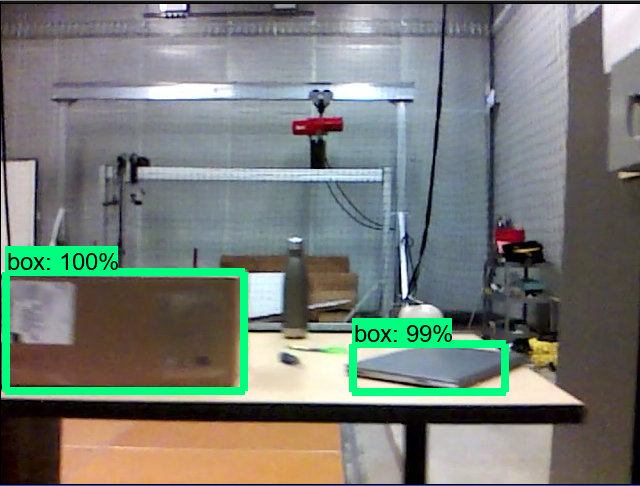
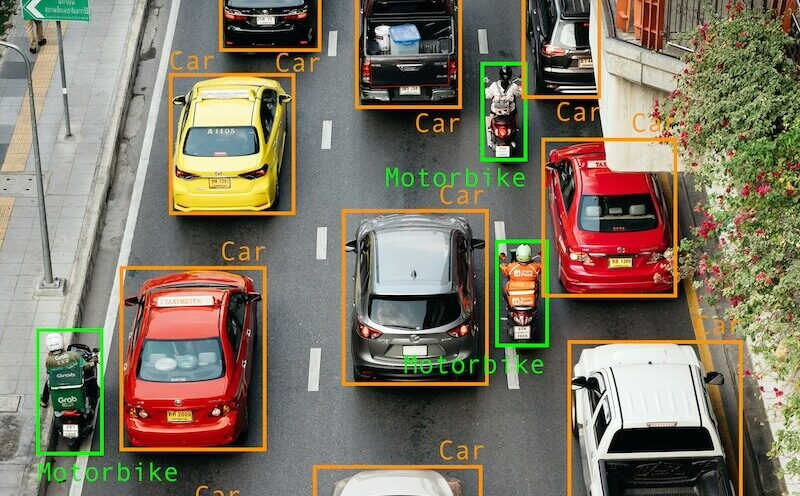
Hayden Wood, Jade Sanchez, Joshua Herold

Cheat Sheet

**Key Concepts**

* **Bounding Box:** Rectangular frames that enclose objects in an image. Represented as coordinates (x, y, width, height).
* **Annotations:** Labels or metadata specifying the class of each object and its location (bounding box).
* **Confidence Score:** Measure the probability that a detected object belongs to a certain class.
* **Intersection over Union (IoU):** Evaluates the overlap between the predicted and ground-truth bounding boxes.



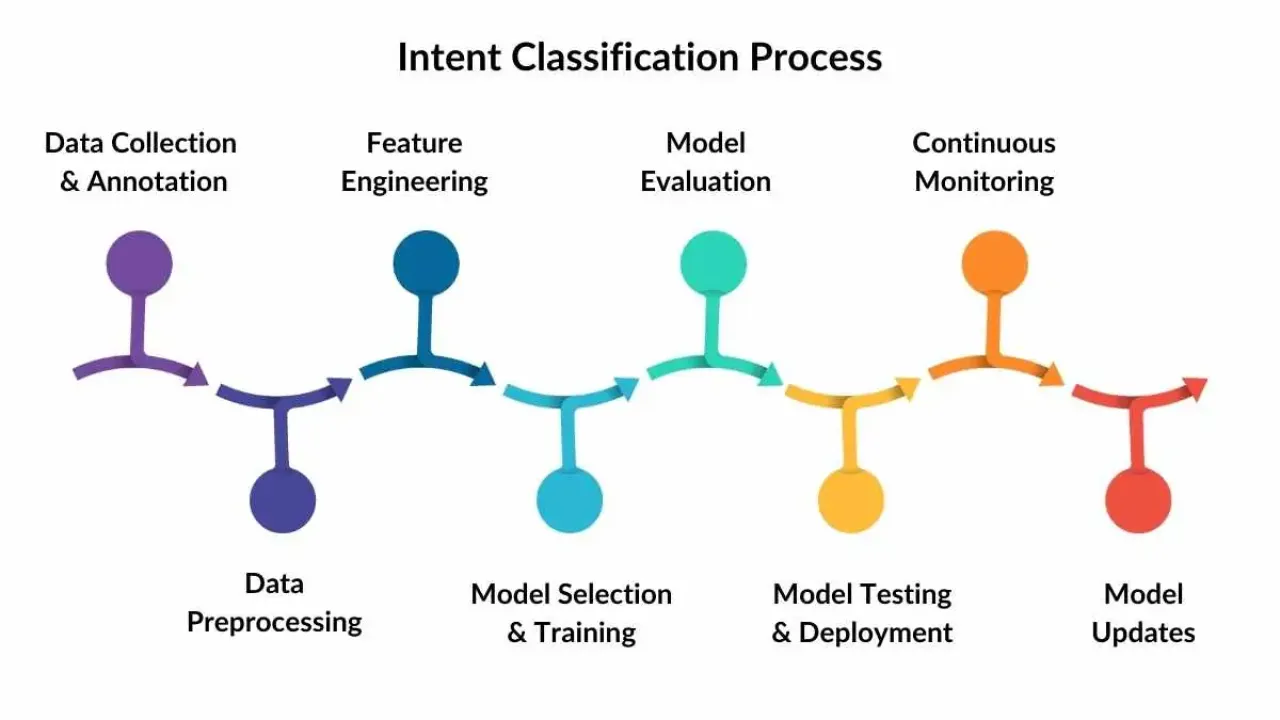


**Algorithms Used in Object Detection**

* **R-CNN (Regions with Convolutional Neural Networks):** Extracts region proposals and applies CNN to each proposal.
* **Fast R-CNN:** Improves R-CNN by using a single network to simultaneously predict bounding boxes and object classes.
* **Faster R-CNN:** Uses a Region Proposal Network (RPN) to generate regional proposals faster than Fast R-CNN. (About 2 to 3 times faster)
* **RPN (Region Proposal Network):** A Faster R-CNN feature that generates region proposals by predicting object limits and objectness scores at each place in an image.
* **SSD (Single Shot MultiBox Detector):** Detects objects in images in a single pass without a region proposal step.
* **YOLO (You Only Look Once):** Divides the image into grid cells and predicts bounding boxes and class probabilities in a single step. Real-time analytics with imprecise results.

**Object Recognition Steps**

1. **Data Collection and Annotation:** Get a set of images for training, validation, and testing.
2. **Data Preprocessing:** Label images, scale pixel values, and use transformations like flipping or rotation to increase variety.
3. **Model Selection:** Choose a suitable algorithm, such as YOLO, SSD, or Faster R-CNN, for object detection.
4. **Training:** Train the model on preprocessed data using deep learning frameworks.
5. **Evaluation:** Use metrics like mAP (mean Average Precision) and IoU to assess model performance.
6. **Inference:** Deploy the trained model for object detection on new images.



**Troubleshooting Tips**

* **High False Positives:** Try lowering the confidence threshold so the model is less likely to detect objects that aren’t really there, or use a more complex model to improve accuracy.
* **Poor Bounding Box Localization:** Increase the IoU threshold during training so the model learns to place boxes more precisely, or make sure your labeled data (annotations) is accurate.
* **Overfitting:** Use techniques like data augmentation (adding variations to training data) and regularization (methods that prevent the model from learning too specific patterns).
* **Imbalanced Classes:** Balance the training data by using techniques like weighted loss functions (giving more importance to rare objects) or resampling (adding more instances of less common classes).

**Additional Resources**

* **Youtube**:
  + <https://youtu.be/nJzQDpppFj0?si=TrQGK0DwkAz61wcT>
  + <https://youtu.be/X3IlbjQs190?si=uLBf78-qjlHU1alB>
  + <https://youtu.be/BRPdiuWO6sA?si=H4UQqQjR2vEm269v>
  + <https://youtu.be/ag3DLKsl2vk?si=e6a4Iih6dy06NK3Z>
  + <https://youtu.be/PlXE1_FVtMQ?si=_pA3HamXBQ4RBLOk>
* **Websites:** 
  + <https://viso.ai/deep-learning/object-detection/#:~:text=Popular%20algorithms%20used%20to%20perform,the%20single%2Dshot%20detector%20family>.
  + <https://www.geeksforgeeks.org/r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-ml/#>
  + <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>
  + <https://www.datacamp.com/blog/machine-learning-lifecycle-explained>

**Reflection**

This assignment gave us a much clearer understanding of how object detection works, including essential concepts like bounding boxes, annotations, confidence scores, and IoU. Diving into different algorithms such as R-CNN, Fast R-CNN, Faster R-CNN, SSD, and YOLO gave us insight into how these methods differ and what makes each one suitable for certain tasks. We also learned how important it is to handle data preprocessing well, from labeling images accurately to using techniques like data augmentation to make our models more robust.

Putting together the cheat sheet was especially helpful because it forced us to break down complex topics into simpler terms, creating a go-to reference that we can easily use later on. Identifying common challenges and figuring out practical solutions also sharpened our troubleshooting skills. We’re confident that this cheat sheet will be a quick guide for future projects, helping us stay organized and better prepared for tackling real-world problems in object detection.