**CSK-Detector: Commonsense in Object Detection**

**Algorithm 1:** CSK-Detector Processing

Input: Images in S, Quantifier Limit L per object

1. Pass images through Mask-RCNN
2. Return basic detected objects from Mask-RCNN
3. For each object O, find quantifier Q
4. If Q > L, return "none: not relevant"
5. Else calculate multifaceted scores: Dice & Eq.3
6. Build training data: objects, scores, intermediate class
7. Pass training data to Decision Tree to learn hypothesis H
8. Use H, CSK-Premises P to trace final image category

Output: Image Category

**What I Have Done with the Gardening Image Dataset (Classes: Kids, Toys, Caregiver, Food)**

1. *Image Input (S):*

I gathered different dataset (tree, fence, outdoor furniture, grass) and modified the data to be in the COCO format needed to model.

<https://universe.roboflow.com/multi-family-common-areas-hb2ji/outdoor-rooftop-object-detection>

<https://universe.roboflow.com/project-fwcli/fence-1zapl>

<https://universe.roboflow.com/olen/tree-detection-dv5i2>

<https://universe.roboflow.com/kmitl-r9ttd/lawn_-swap>

<https://universe.roboflow.com/mower-gyzd2/mower-2r1ou>

1. *Pass Images through Mask-RCNN (Step 1):*

I used a Mask-RCNN model to process the Garden image dataset, responsible for detecting various objects within the images, such as

['barstools', 'bush', 'chair', 'dog', 'fence', 'firepit', 'furniture', 'human', 'outdoor\_accent\_chair', 'outdoor\_chaise', 'outdoor\_coffee\_table', 'outdoor\_dining\_chairs', 'outdoor\_dining\_table', 'outdoor\_end\_tables', 'outdoor\_sofa', 'plants', 'pole', 'rock', 'sprinkler', 'trash', 'tree', 'umbrella']

1. *Return Basic Detected Objects (Step 2):*

After running Mask-RCNN, I retrieved the custom detected objects in each image.

1. *Quantifier Calculation (Step 3) (For Each Object O, Find Quantifier Q):*

For each object detected by Mask-RCNN, I calculated a quantifier (Q), which refers to the count of each detected object in an image.

1. *Quantifier Comparison (Step 4) (If Q > L, Return "None: Not Relevant"):*

I established a limit (L) for the number of occurrences of each object. If the detected number (Q) exceeded this limit, the image was labeled as "not relevant" for the Nursery category.

1. *Multifaceted Scores Calculation (Step 5) (Else Calculate Multifaceted Scores: Dice & Eq.3):*

For objects that did not exceed the limit, I calculated multifaceted scores (e.g., Dice score, Eq.3). These scores are likely related to the object’s relevance, similarity, or classification in the context of nursery objects.

1. *Build Training Data (Step 6): (image\_detection\_results\_with\_labels.csv)*

Using the detected objects, scores, and intermediate class labels, I built a dataset for training a Decision Tree. This dataset included:

* + Detected objects
  + Multifaceted scores
  + Intermediate class labels (e.g., classification categories like "Garden" or "Not Garden")

1. *Train the Decision Tree (Step 7):*

The training data was fed into a Decision Tree classifier to learn the hypothesis (H) that maps objects and scores to the final image category.

1. *Final Image Category (Step 8):*

Using the learned hypothesis (H) and commonsense knowledge (CSK-Premises P), I traced the final image category, which was either "Garden" or "Not Garden."

**Testing and Comparison Summary:**

1. *Testing Process:*

The testing process involved evaluating the system's ability to correctly categorize images as "Garden" or "Not Garden" based on detected objects and scores.

Refer: *(test\_predictions.csv) for testing image predictions using CSK*

1. *Testing Metrics:*
   * Accuracy: I calculated the accuracy of the Decision Tree and possibly other models (like CLIP) to determine how well the classifier performed.
   * Comparison with CLIP Model: I have tested the CLIP model for comparison, evaluating how well CLIP can classify the same images compared to the Decision Tree model.

Refer: (*clip\_predictions.csv) for testing image predictions using CLIP*

1. *Summary of Results:*

o CSK Detector Performance: Based on testing, this achieved an accuracy of 0.88, indicating it classified 88% of the images correctly. It demonstrated high recall (0.88), and a strong F1-score (0.94).

o CLIP Model Comparison: The CLIP model achieved an accuracy of 0.67, lower than the CSK Detector model. It has lower recall (0.67) and F1-score (0.80) compared to the above

Refer: (comparison\_predictions.csv) for comparison