Group 3 – "Deteção de Lixo no Chão"

You already have a clean TACO→YOLO pipeline, a YOLOv8-nano baseline, and first mAP ≈ 0.63 on five super-classes. For next steps you need to (a) lift the glass/other performance, (b) reduce tiny-object false-positives, (c) show you explored at least two axes of the "architecture × data × training" design space—without exceeding one free-tier Colab week.

Below is a **7-day, ≤ 6 GPU-h roadmap**.

---

0 One-time fixes to lock in on Day 1

Fix	How	Why
Class-imbalance weights	In `yolo train` add `--cls_weights 1.0 1.0 1.0 2.0 2.5` (glass, other heavier).	Balances BCE part of loss; almost free.
Small-object anchor tuning	`--anchors 3` (NA = 3) + `--rect` flag; repicks anchor sizes on your data.	YOLO default anchors are COCO-centric (median obj ≈ 10 % image).
Mosaic & Copy-Paste aug	Keep Ultralytics defaults **on**, but raise `--degrees 10 --scale 0.5 1.5`.	Larger scale range → glass fragments appear bigger; copy-paste boosts rare classes.
Label-sanity script	Plot 200 random bboxes; hand-check for mis-labels after your superclass merge.	Wrong targets doom glass & other classes.

---

1 Experiment grid – pick three (Days 2-5)

ID	Hypothesis	Change vs. baseline	Expected Δ	GPU h
A	Bigger feature maps help small trash	**YOLOv8-s** (11 M params) @ **imgsz = 768**; batch = 8, AMP.	↑ mAP (glass + other) 4–6 pp	1.5
B	Focal loss reduces BG confusion	`--fl_gamma 2.0` (focal) + `--cls_weights` as above.	↓ FP small shards, ↑ precision	0.3
C	Synthetic oversampling boosts rare classes	Run **Albumentations Copy-Paste** script to replicate 1 k glass/other instances → new train folder, then retrain nano 50 ep.	↑ AP_glass 6 pp, other 4 pp	0.8
D	Segmentation pre-training transfers better	Fine-tune **YOLOv8-seg-s** (weights from COCO-seg) on TACO masks 40 ep → export only detector head.	↑ box AP via mask cues	1.5
E	Multi-dataset pre-train improves generalisation	1. Fine-tune on **TrashNet + pLitterStreet** (5 ep) → 2. Continue on TACO (50 ep).	↑ recall on shiny glass & wrappers	1.2

ID	Hypothesis	Change vs. baseline	Expected Δ	GPU h
F	Post-process size filter cuts tiny FP	After NMS, drop boxes with `wh < 0.02·imgsz`.	↑ precision 3 pp, negligible recall loss	0

2Training hygiene

- **Mixed precision** (`--device 0 --half`) saves 40 % VRAM.
- **Grad-acc 2** keeps effective batch = 16 even if memory allows 8.
- **Early-stop** patience = 10 on `metrics/mAP50` to avoid 100-epoch over-train.
- Log to **Weights & Biases** (already integrated in Ultralytics) → metrics survive Colab resets.

---

3 Evaluation protocol (fixed for all runs)

1. **Split** – 5-fold stratified by superclass counts (you already have train/val/test; put test aside).
2. **Metrics** – `mAP50` (overall), per-class AP, precision/recall at conf = 0.25, FPS@T4.
3. **Stat test** – Bootstrap 1000× AP_glass/other vs. baseline; report CI.
4. **Qualitative** – 10-image collage of successes + top-5 FP/FN crops for glass & other.

---

4 One-week timeline (GPU ≈ 6 h)

Day	Agenda
1	Anchor-tune, class weights, sanity plots; re-run nano 20 ep → new *Baseline-v2*.
2	Train **YOLOv8-s@768** (Exp A).
3	Train **nano+Focal** (Exp B) **and** run copy-paste script.
4	Retrain **nano+synthetic** (Exp C).
5	Optional Exp E or segmentation pre-train (Exp D) if GPU budget remains.
6	Aggregate metrics across folds; size-filter experiment; create Grad-CAMs.
7	Write Milestone-2: scoreboard, PR curves, compute budget, lesson-learned paragraph.

---

5 Scoreboard template for the report

Exp	Model / imgsz	AP50 ↑	AP_Plast ↑	AP_Paper ↑	AP_Glass ↑	AP_Other ↑	GPU min
Base-v2	v8-n / 640	63.3	72.1	68.4	41.0	38.2	30
A	v8-s / 768	**67.8**	74.3	70.2	**49.5**	43.7	55
B	v8-n / 640 + Focal	65.4	73.2	69.1	46.2	41.9	35

Exp	Model / imgsz	AP50 ↑	AP_Plast ↑	AP_Paper ↑	AP_Glass ↑	AP_Other ↑	GPU min
C	v8-n / 640 + synth	66.6	72.8	**71.4**	47.7	**46.3**	40

Shade column bests; mark "✓" in write-up when 95 %-CI excludes baseline.

Quick-wins checklist

- **Anchor-tune** + class weights
- **YOLOv8-s @ 768** for small litter
- **Focal loss** to curb BG confusion
- **Copy-paste** to oversample glass/other
- **Size-filter** post-NMS tiny boxes
- Report per-class AP & bootstrap CI

Follow this plan and you'll deliver a report that clearly shows *which* choices moved the needle for the hardest litter classes—while staying well inside the free Colab compute budget. Good luck!