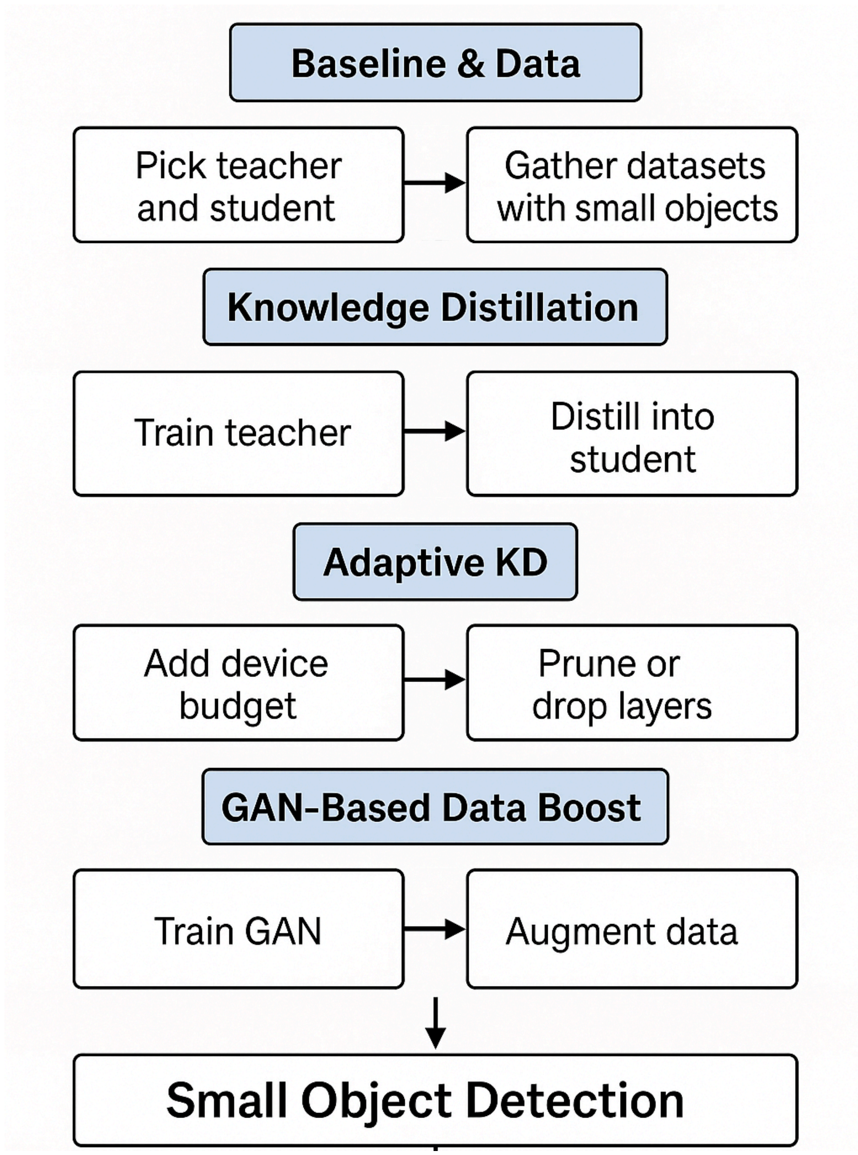
Project Outline

Title: GAN-Driven Adaptive Knowledge Distillation Framework for Small Object Detection



1. Goal

Build a light-weight detector that spots tiny objects (think drones, road signs) on an edge device without losing accuracy.
We do this in three stages—distil, adapt, and augment—and later test an anchor-free head for extra gains.

---

2. Step-by-Step Plan

Stage	What We Do	Why It Matters	Output
A. Baseline & Data	• Pick *teacher* (e.g., Faster R-CNN-ResNet50) and *student* (e.g., MobileNet-SSD). • Gather datasets with many small objects: COCO-small subset, VisDrone.	Gives us starting scores and a fair yard-stick.	Cleaned datasets + baseline mAP, FPS, model size.
B. Plain Knowledge Distillation (KD)	• Train teacher to high accuracy.• Distil teacher "soft labels" into student.	Shrinks the model while copying smarts from teacher.	Student-KD model; compare with baseline.
C. *Adaptive* KD	• Add a device budget (RAM, latency).• Use pruning or layer-drop guided by that budget **during KD** (e.g., reward fast layers).	Makes the student suit each device (phone vs. Jetson).	Multiple student variants + speed/accuracy table.
D. GAN-Based Data Boost	• Train a lightweight GAN offline to create sharper or zoomed-in versions of small objects.• Mix real and GAN images when re-training student.	More varied tiny objects → better recall. GAN runs only in training, so runtime cost = 0.	Augmented dataset + new student scores.
E. Anchor-Free Head (Exploratory)	• Replace SSD head with FCOS/NanoDet head.• Re-apply Adaptive KD on the new head.	Anchor-free detectors often suit small objects.	Comparison chart: anchor vs. anchor-free.

F. Evaluation & Ablation	• Measure **mAP-small**, FPS, model size on edge device.• Ablate: KD only, KD+Adapt, KD+Adapt+GAN, +Anchor-free.	Shows which part adds real value.	Final report & charts.

---

3. Tentative Timeline (12 weeks draft)

1. **Week 1–2** Data prep + baseline models.
2. **Week 3–4** Train teacher & plain KD student.
3. **Week 5–6** Implement adaptive KD, profile on target device.
4. **Week 7–8** Train GAN, create augmented images, retrain student.
5. **Week 9** Anchor-free head integration (optional stretch).
6. **Week 10** Full evaluation + ablation studies.
7. **Week 11** Write paper / slides, polish code repo.
8. **Week 12** Dry-run demo & final presentation.

---

4. Resources Needed

- **Compute:** 1 GPU (>= 12 GB) for teacher & GAN training; CPU/Jetson Nano for on-device tests.
- **Libraries:** PyTorch, torchvision, albumentations, Detectron2 / MMDetection (for FCOS), GAN module (e.g., CycleGAN or ESRGAN-lite).
- **Storage:** ~200 GB for datasets + checkpoints.

---

5. Success Metrics

Metric	Target
mAP-small ↑	+5 points over baseline SSD.
Model size ↓	≤ 25 MB (compressed).
Latency	≤ 30 ms per frame on target device.
Ablation proof	Each added module shows ≥ 1 mAP gain.

---

6. Risks & Mitigation

Risk	Mitigation
GAN overfits / adds artifacts	Use moderate augment ratio; validate images manually.
Adaptive KD hurts accuracy	Tune pruning threshold; fall back to non-pruned student for that device.
Time crunch on anchor-free head	Treat as optional stretch; core deliverable stops at Stage D.

---

7. Final Deliverables

1. **Code repo** (clean, documented, reproducible).
2. **Trained models** (teacher, student variants).
3. **Technical report** (≤ 8 pages) with results & ablations.
4. **Demo video** running on edge device.
5. **Slide deck** for presentation.