Unified-OneHead Multi-Task Challenge

*Object Detection + Semantic Segmentation + Image Classification with a **single head***

---

0. TL;DR

Key Point	Requirement
Task	Build **one head (2–3 layers)** that outputs detection + segmentation + classification simultaneously
Data	Three *mutually exclusive* mini-datasets (< 120 MB total) supplied by the course; one-command download
Hardware	Free Google Colab GPU (T4 or V100) • **Total training time ≤ 2 h**
Core difficulty	*Catastrophic forgetting*: after alternating training, each task must drop ≤ 5 % vs. its single-task baseline
Deliverables	`colab.ipynb` • `report.md` • llm_dialogs.zip (all LLM chat logs)
Grading	20 + 20 + 25(+5) + 10 + 15 + 10 = 100 pts
Deadline	**XX/YY (Fri) 23:59 GMT+8** — submit via GitHub pull request

---

1. Official Mini-Datasets

Task	Subset Name	Origin	Download Size	Images (train/val)	Annotation
Detection	**Mini-COCO-Det**	COCO 2017 (10 classes)	45 MB	300 (240 / 60)	COCO JSON
Segmentation	**Mini-VOC-Seg**	PASCAL VOC 2012	30 MB	300 (240 / 60)	PNG masks
Classification	**Imagenette-160**	Imagenette v2	25 MB	300 (240 / 60)	folder / label

All image files are
non-overlapping across the three subsets.Hidden test sets (60 images per task) will be used by the TAs for final evaluation.

Dataset directory layout after extraction:

```
data/
├── mini_coco_det/{train,val}
├── mini_voc_seg/{train,val}
└── imagenette_160/{train,val}
```

---

2. Model Specifications

Component	Constraint
Backbone	Choose **one**: MobileNetV3-Small (2.5 M), EfficientNet-B0 (5.3 M), YOLOv8-n *backbone* (3.2 M), or Fast-SCNN (1.1 M). Public ImageNet weights may be loaded.
Neck	≤ 2 conv/BN/ReLU layers *or* a single FPN layer.
Head	Exactly **2–3 layers** (Conv or MLP) and **single branch**: must emit *all three* outputs at once. Example output schema (stride 16 feature map): • Detection: N × (cx, cy, w, h, conf, C_det) • Segmentation: C_seg × H × W mask • Classification: C_cls logits
Total parameters	**< 8 M** (entire model).
Inference speed	≤ 150 ms per 512 × 512 image on Colab T4.

---

3. Training Schedule & Forgetting Criterion

```
Stage 0 (optional) — warm-up / ImageNet pretrain
Stage 1 — Train ONLY on Dataset A (Seg)  →  record mIoU_base
Stage 2 — Train ONLY on Dataset B (Det)  →  measure mIoU_drop
Stage 3 — Train ONLY on Dataset C (Cls)  →  measure mIoU_drop +
mAP_drop
```

Required after Stage 3:

```
mIoU ≥ mIoU_base – 5 %
mAP  ≥ mAP_base  – 5 %
Top-1 ≥ Top1_base – 5 %
```

3.1 Forgetting-mitigation toolbox (pick any)

- Elastic Weight Consolidation (EWC)
- Learning without Forgetting (LwF)
- Replay buffer (≤ 10 images per task per step)
- Knowledge distillation (previous checkpoint as teacher)

---

4. Scoring (100 pts)

Item	Description	Pts
Design & motivation	Soundness and creativity of single-head architecture	20
Training schedule & forgetting remedy	Completeness and theoretical justification	20
Performance	All tasks within the 5 % drop → 25 pts *Bonus + 5 pts* if **every** metric ≥ its baseline	25 (+5)
Resource efficiency	Training ≤ 2 h, params < 8 M, inference < 150 ms	10
Report quality	Clarity, figures, citations	15
LLM dialogue logs	Complete prompts + responses	10

---

5. Deliverables

```
your_repo/
 ├─ colab.ipynb        ← runnable end-to-end notebook
 ├─ report.md          ← architecture, schedule, results, analysis
 ├─ llm_dialogs.zip    ← ALL ChatGPT / LLM logs (txt or json)
 ├─ README.md          ← reproduction steps & requirements
 └─ scripts/           ← helper & evaluation scripts
```

Submit a GitHub pull request to the course repo including the commit hash.

---

6. Evaluation Command (used by TAs)

```
python eval.py \
  --weights your_model.pt \
  --data_root data \
  --tasks all          # outputs mAP, mIoU, Top-1
```

7. FAQ

Question	Answer
Can I use a different backbone?	Yes, as long as the *total* parameter count is < 8 M and you justify your choice in the report.
May I implement the head as a small Transformer?	Yes—≤ 3 layers, still within the parameter limit.
How do I balance the three losses?	Up to you; describe your search strategy in the report.
Are LLM logs really mandatory?	Yes. Missing or incomplete logs = 0 pts for that category.
Can I train on multiple GPUs or paid Colab?	No. The TA validation uses a single free T4/V100. Your code must reproduce results in that environment.

8. Academic Integrity

- **Provide all LLM conversations** in `llm_dialogs.zip`.
- External code/data must be cited.
- Plagiarism or undisclosed extra data leads to point deductions or failure.