Computer Vision Capstone Projects

AIMS Cameroon · Structured Master's, Data Science · 2026–2027

Released: Week 2, Day 5 · Presentations: last day of Week 3

Instructor: Yusuf Brima, Ph.D.

1 What the capstone is

In teams of **2–3**, you will apply what you learned in Weeks 1–3 to a real computer vision problem: frame it, obtain and check the data, build a baseline, improve on it, and evaluate the result honestly, including where it fails. You may choose one of the twelve proposed topics below or propose your own.

The goal is *not* a new architecture. A good capstone is **small, real and honestly evaluated**: a clear question, a credible dataset, a baseline, a stronger model, a test set that reflects how the system would really be used, and an analysis of failures.

2 Timeline

When	What
Week 2, Day 5	Topics released (this document). Self-proposed projects: submit the one-page synopsis (Section 5) by the end of the day.
Week 3, Day 1	Teams and topics confirmed. Every team has downloaded its data and looked at it.
Week 3, Days 2–3	Baseline running by Day 2; main model by Day 3. Consultations with the instructor (each team books a slot).
Week 3, Day 4	Evaluation, failure analysis, slides, optional demo video. Code frozen by the evening.
Week 3, Day 5	Final presentations.

3 Deliverables

1. **Presentation** on the last day of Week 3: about 10 minutes plus 5 minutes of questions, every team member speaking. A short demo video (at most 2 minutes) may be included within the time.
2. **Code**: a repository or notebook that reproduces your main numbers from the data, with a README explaining how to run it and where the data come from.
3. **One-page project summary** (LaTeX, PDF): question, data, method, main result table, two failure examples, limitations, and each member's contribution.

Instructor note: adjust presentation length and the summary requirement to the final number of teams.

4 How projects are assessed

Criterion	Weight
Problem framing: who is the user, what decision does the output support, which errors matter	15%
Data: source and licence, labelling quality, size, *splitting without leakage* (by site, patient, session or time)	20%
Methods: a sensible baseline and a justified main model, using course tools correctly	20%
Evaluation: appropriate metrics, per-class results, a test set that reflects deployment, uncertainty where relevant	20%
Failure analysis and limitations: examples, explanations (e.g. Grad-CAM), what you would do next	15%
Presentation and reproducibility: clear story, readable figures, code that runs	10%

A project whose model is only modestly better than its baseline, but which is evaluated honestly and explains why, will score higher than an impressive number from a leaky split.

5 Proposing your own project

Submit a one-page synopsis by the end of Week 2, Day 5. It must contain:

1. **Problem statement**: what you want the system to do, in one or two sentences.
2. **Motivation**: who would use it, and why it matters (in Cameroon or elsewhere).
3. **Proposed methodology**: the task type (classification, detection, segmentation, ...), a baseline, and the main model.
4. **Data access plan**: where the data come from, their licence, how many images, how they are or will be labelled, and how you will split them. *Projects without a credible data plan will not be approved.*
5. **Success criterion**: the metric and the level that would make the result useful.

The example synopsis on the last slides of Week 2, Day 5 shows the expected level of detail.

6 Practical rules

- **Compute**: everything must be feasible on a free Google Colab or Kaggle GPU within a few hours of training. Prefer pre-trained models and fine-tuning (Week 2, Day 2).
- **Data ethics**: images of people, number plates, homes or patients are personal data. Ask consent when you photograph people, blur faces and plates in anything you show, and use only public, de-identified medical datasets. Respect dataset licences and cite every dataset and model you use.
- **Own photos**: several topics ask you to collect a small local test set. Thirty to sixty carefully labelled local images are worth more than thousands of extra training images from elsewhere, because they show whether your model works *here*.
- **AI assistants**: allowed as for assignments. Say what you used them for; every team member must be able to explain all the code.

7 The twelve proposed topics

	Topic	Task	Area
P1	Cassava disease diagnosis from phone photos	classification	agriculture
P2	Measuring leaf-disease severity	segmentation	agriculture
P3	Malaria parasite screening in blood smears	classification / detection	health
P4	Tuberculosis screening that generalises across hospitals	classification	health
P5	Pothole and road-damage detection	detection / segmentation	infrastructure
P6	Counting vehicles and moto-taxis from street video	detection + tracking	transport
P7	Land-cover mapping from Sentinel-2, tested on Cameroon	classification	earth observation
P8	Flood mapping from satellite images	segmentation	earth observation
P9	Wildlife camera-trap triage	classification	conservation
P10	CFA franc banknote recognition for blind users	classification (on-device)	accessibility
P11	Grading produce ripeness at the market	classification / detection	agriculture
P12	Helmet-use monitoring with a label-efficient data engine	detection	road safety

Each topic below lists a *minimum* deliverable (enough for a good mark if done well) and *stretch* goals. Dataset sizes and licences were checked when this document was written; confirm them before you start.

P1 Cassava disease diagnosis from phone photos

Why it matters. Cassava is a staple food for hundreds of millions of people in Africa, including Cameroon. Cassava mosaic disease and brown streak disease can destroy much of a harvest, and there are far too few extension officers to inspect every field.

Data. **iCassava 2019** (Makerere AI Lab, Uganda): about 9 400 labelled phone photos in 5 classes (four diseases and healthy), already split into training, validation and test sets in TensorFlow Datasets (`cassava`). A larger Kaggle version (about 21 000 images) needs a Kaggle account. *Plus*: at least 40 photos of cassava leaves you take yourselves, labelled with help from someone who knows the diseases if possible.

Minimum. Frozen ResNet/ConvNeXt features + logistic regression (baseline), then fine-tuning (Week 2, Day 2). Per-class precision and recall, confusion matrix, Grad-CAM on correct and incorrect predictions.

Stretch. Test on your own local photos and quantify the drop; data-efficiency curve (as for beans); a small model (MobileNet) timed on a phone or CPU; test-time augmentation.

Pitfalls. Class imbalance (one disease dominates); near-duplicate photos of the same plant; label noise between visually similar diseases.

P2 Measuring leaf-disease severity

Why it matters. Agronomists decide on treatment from *severity*, the percentage of leaf area affected, not just "diseased or not". Estimating it by eye is slow and inconsistent.

Data. Leaf photos from iBean (Hugging Face `AI-Lab-Makerere/beans`, MIT licence) or iCassava, plus your own photos. You create **masks** for about 60–100 leaves (leaf and lesion classes) with CVAT or Label Studio, using SAM-assisted labelling (Week 2, Day 4).

Minimum. Classical baseline: the Week 1 HSV colour rule for lesions. Main model: U-Net with a pre-trained encoder. Report IoU and Dice per class and the error in predicted severity (percentage points) against your labels.

Stretch. Measure agreement between two team members labelling the same 20 leaves (your ceiling); a severity scale (e.g. 0–5) and its accuracy; failure analysis on shadows and dry leaf edges.

Pitfalls. Inconsistent masks; JPEG-saved masks (use PNG); splitting different photos of the same leaf across train and test.

P3 Malaria parasite screening in blood smears

Why it matters. Malaria is endemic in Cameroon. Microscopy remains the reference test, but reading smears is slow and depends on scarce expert microscopists.

Data. **Level 1**: NIH/NLM malaria cell images: 27 558 single-cell crops (parasitised or uninfected) from thin smears (public download from `data.lhncbc.nlm.nih.gov`). **Level 2**: whole-slide-field detection with bounding boxes, e.g. the Lacuna Malaria Detection data (Uganda; Zindi account required).

Minimum. Level 1: small CNN from scratch vs. fine-tuned pre-trained model; report sensitivity (recall of parasitised cells) and specificity, and choose a threshold for screening (missed parasites are costly).

Stretch. Level 2: fine-tune a YOLO detector; report AP and count accuracy per image; compare with the Week 1 colour-rule detector; the NIH data come from several patients: split by patient if the metadata allow, and explain why it matters.

Pitfalls. Patient leakage between train and test; reporting accuracy on a balanced set when real smears are mostly negative.

P4 Tuberculosis screening that generalises across hospitals

Why it matters. Tuberculosis is a leading infectious cause of death worldwide, and many regions lack radiologists. Models that work in one hospital often fail in another, because scanners and patient populations differ.

Data. Two public chest X-ray sets from NLM: **Shenzhen** (China, 662 images, normal/TB) and **Montgomery County** (USA, 138 images). Both de-identified and publicly available.

Minimum. Fine-tune a pre-trained CNN on one hospital's images and test on the *other* hospital's, and vice versa. Report sensitivity, specificity and AUC in both directions, and compare with a within-hospital split.

Stretch. Grad-CAM: does the model look at the lungs or at text markers and borders? Mask out everything except the lungs and retrain; preprocessing (CLAHE, Week 1, Day 3) as a domain-shift remedy.

Pitfalls. Very small data: use strong pre-training, augmentation and cross-validation. Never present such a model as a clinical tool.

P5 Pothole and road-damage detection

Why it matters. Road damage causes accidents and vehicle wear. Cities and road agencies need to know where damage is, and how it changes, without driving every road with an inspector.

Data. Public pothole datasets (e.g. the Roboflow pothole segmentation set on Hugging Face, `keremberke/pothole-segmentation`; RDD2022 for multi-class road damage). *Plus*: at least 50 photos you take on roads in Yaoundé or elsewhere, with boxes or masks, as a local test set.

Minimum. Fine-tune a small YOLO (detection or segmentation). Report AP50 and AP on the public test set *and* on your local photos; explain the difference.

Stretch. Estimate damaged area from masks; a map of detections from geotagged photos; confidence-threshold choice for an inspection workflow.

Pitfalls. Puddles and shadows that look like potholes; different camera heights and angles between datasets.

P6 Counting vehicles and moto-taxis from street video

Why it matters. Traffic counts guide road planning and public transport. In many African cities, moto-taxis are a large share of traffic but missing from standard detector classes.

Data. Short videos you record at a junction (a few minutes from a fixed, safe position, with permission where needed). Label a sample of frames with boxes for *car*, *taxi*, *motorcycle*, *bus/truck*, *person*.

Minimum. Pre-trained YOLO + a tracker (ByteTrack in Ultralytics) to count vehicles crossing a line. Compare automatic counts with manual counts on at least 3 short clips; report count error per class.

Stretch. Fine-tune on your labelled frames to separate taxis and moto-taxis from other vehicles; occlusion and night analysis; frames per second on a CPU.

Pitfalls. Privacy: blur faces and plates in anything you present; identity switches in crowded traffic; counting the same vehicle twice.

P7 Land-cover mapping from Sentinel-2, tested on Cameroon

Why it matters. Mapping forest, crops, settlements and water over time supports planning, conservation and climate adaptation. Models trained on European imagery may not transfer to Central African landscapes.

Data. **EuroSAT** (27 000 Sentinel-2 patches, 10 classes; RGB and 13-band versions on Zenodo and Hugging Face). *Plus*: at least 100 patches you cut from free Sentinel-2 images of Cameroon (Copernicus Browser) and label yourselves with the classes that apply.

Minimum. Fine-tune a pre-trained CNN on EuroSAT RGB; test on your Cameroon patches; report per-class results and the size of the domain gap.

Stretch. Use all 13 bands by adapting the first layer (Week 2, Day 2); an NDVI-threshold baseline for vegetation (Week 1, Day 2); fine-tune on a few labelled Cameroon patches and plot accuracy against their number.

Pitfalls. Classes that do not exist in Cameroon (and vice versa); cloud cover; patches from the same scene in both training and test sets.

P8 Flood mapping from satellite images

Why it matters. Floods affect parts of Cameroon and the region each rainy season. Rapid maps of flooded areas guide emergency response.

Data. **Sen1Floods11**: Sentinel-1 radar and Sentinel-2 optical chips with hand-labelled water masks, from flood events on several continents including Ghana, Nigeria and Somalia (public, Google Cloud bucket; see the project's GitHub page).

Minimum. Baseline: a water-index threshold (NDWI from Sentinel-2 bands). Main model: U-Net on Sentinel-2 (and/or Sentinel-1). Report IoU of the water class per region.

Stretch. Train on some flood events and test on held-out ones (especially the African events); compare radar and optical inputs (radar sees through clouds); a flood map for one real event.

Pitfalls. Cloud-covered optical chips; permanent water vs. flood water; splitting chips from the same event across train and test.

P9 Wildlife camera-trap triage

Why it matters. Camera traps produce millions of images, most of them empty (wind, grass). Ecologists spend weeks sorting them. Automatic triage frees them to study the animals.

Data. A subset of **Snapshot Serengeti** (Tanzania), hosted on LILA BC (`lila.science`), with labels for empty images and species. Choose 5–10 species plus "empty" and download a few thousand images using the metadata files.

Minimum. Step 1: empty vs. animal classifier. Step 2: species classifier on the non-empty images. Split by **camera location**, report per-class recall, and estimate how much manual work the system saves at a chosen threshold.

Stretch. Compare with an animal detector (e.g. MegaDetector) as a first stage; night (infrared) vs. day performance; rare-species recall.

Pitfalls. Images from the same camera and burst in train and test (the most common leak in this field); extreme class imbalance.

P10 CFA franc banknote recognition for blind users

Why it matters. People with visual impairment need to identify money independently. An offline phone model for Central African CFA franc (BEAC) notes and coins is a concrete accessibility tool.

Data. You collect it: every current banknote denomination (and optionally coins), at least 40 photos each, front and back, under varied light, backgrounds, folds and partial occlusion by fingers. Keep the photos from one or two sessions apart as a test set.

Minimum. A MobileNet or EfficientNet fine-tuned for classification; per-denomination accuracy on the held-out session; measured latency on a phone or CPU.

Stretch. An "unsure" option when confidence is low (calibration, Week 2, Day 1); robustness to blur and low light; a simple app or notebook demo with spoken output.

Pitfalls. Backgrounds that differ by denomination (a shortcut); too few photos of worn notes; evaluating on photos from the same session as training.

P11 Grading produce ripeness at the market

Why it matters. Ripeness determines price and waste for plantain, banana, tomato and mango. Consistent grading helps farmers and traders.

Data. You collect it at a market or farm: one or two products, 3–4 ripeness grades defined with a trader or agronomist, at least 50 photos per grade, in market lighting. Optionally boxes for several fruits per image.

Minimum. Baseline: colour features in HSV (Week 1, Day 2) + logistic regression. Main model: fine-tuned CNN. Compare, with a confusion matrix: are errors only between neighbouring grades?

Stretch. Detection of several fruits per photo and per-fruit grading; agreement between two human graders as a ceiling; robustness to lighting colour.

Pitfalls. Grades that are inconsistently defined; one batch of fruit per grade (so the model learns the batch, not ripeness).

P12 Helmet-use monitoring with a label-efficient data engine

Why it matters. Moto-taxi crashes are a major cause of injury in Cameroonian cities, and helmets greatly reduce head injuries. Measuring helmet use helps target road-safety campaigns.

Data. Your own street photos or video frames of motorcycles (with care for privacy: report only anonymous counts, blur faces in anything shown). Classes: *rider with helmet*, *rider without helmet*, *motorcycle*.

Minimum. The data engine of Week 2, Days 4–5: pre-label 100 images with an open-vocabulary detector (OWL-ViT or Grounding DINO) and/or SAM; correct the labels; fine-tune a small YOLO; report AP per class. Measure how much labelling time pre-labelling saved.

Stretch. A second round of the data engine; helmet-use rate estimated from video with tracking; errors for passengers and small, distant riders.

Pitfalls. Helmets carried in hands or hung on handlebars; heavy occlusion between passengers; publishing identifiable images (do not).

8 A checklist for every team

- ☐ We can state the user and the decision our system supports in one sentence.
- ☐ We have looked at at least 50 of our images and their labels by eye.
- ☐ Our test set comes from a different site, session, patient group or time than our training set, and we can explain why.
- ☐ We have a baseline result before we tune the main model.
- ☐ We report per-class results, not only a single average.
- ☐ We show at least two failure cases and a plausible explanation for each.
- ☐ Every dataset and pre-trained model is cited with its licence.
- ☐ Someone outside the team can run our code from the README.