HW: Visual State Detection with Transfer Learning & Gradio

ME 405: Applied Deep Learning

Due: Oct 21 2025 — **Submit on Canvas:** PDF writeup, Colab notebook (with saved outputs), trained model

---

Overview & Motivation

Build a simple computer-vision system that detects a *visual state you care about* and that would be useful to check *remotely* (e.g., safety, energy/water use, convenience). You will collect short videos, construct a reproducible data pipeline in Google Colab, train a transfer-learning classifier, evaluate accuracy, and deploy a working *Gradio* demo usable on a phone.

You choose the task and #classes. Students may choose any binary or multi-class problem (**2 or 3 classes are recommended**; more classes are allowed but will require more data/effort).

Suggestions (examples only — choose your own if you prefer).

- Faucet: `running / off / dripping`
- Door: `open / closed`
- Space heater: `on / off / standby`
- **Parking spot**: `occupied / empty`
- Laundry: `washer running / idle`
- Fridge: `door ajar / closed`
- Device indicator LED: `on / off / blinking`
- 3D printer: `printing / idle`

These are **only suggestions**; pick anything that is useful for you and that a camera can observe reliably.

Learning Objectives

1. Frame a clear visual classification problem with engineering relevance.
2. Build a dataset from your own videos; avoid data leakage via session-wise splits.
3. Apply data augmentation appropriately (domain-aware).
4. Fine-tune a pretrained CNN (e.g., ResNet-18 / MobileNetV2 / EfficientNet-B0).
5. Evaluate with accuracy; analyze failures.
6. Deploy a Gradio app for mobile testing; discuss limitations and ethics.

Task Requirements (What You Must Do)

1) Problem Definition (1–2 paragraphs)

Define your task and the number of classes k (you choose; **2–3 recommended**). State the engineering significance (e.g., water/energy safety, HFE, reliability). Define each class precisely (include edge cases like "*dripping*" vs. "*off*").

2) Data Collection (original videos)

Capture **5–10 minutes total** of video across **multiple angles, distances, and lighting**. Record at least **3 sessions** (different times of day). Avoid faces/bystanders.

3) Data Pipeline (Colab)

Extract frames at 2–4 fps into JPEG and organize them into training/validation/test splits by session (not random frame-level splits). Keep a manifest (CSV) of file paths, labels, and session IDs.

4) Data Augmentation

Apply label-preserving transforms: random crop/resize, small rotations, brightness/contrast jitter, horizontal flip only if meaningful in your domain, and light Gaussian noise. Log which transforms helped or hurt.

5) Model & Training (Transfer Learning)

Use a pretrained backbone (ResNet-18 / MobileNetV2 / EfficientNet-B0). Replace the classifier head for k classes. Strategy:

1. Freeze backbone; train the new head.
2. Unfreeze last block(s); fine-tune with a lower LR.

Targets: Validation accuracy $\geq 90\%$ (well-justified exceptions allowed if you conduct strong failure analysis).

6) Evaluation

Report accuracy and 6–8 failure examples with brief reasoning (lighting, reflections, occlusion, etc.).

7) Deployment (Gradio)

Create a Gradio app that accepts a phone camera snapshot and outputs predicted class + confidence. Provide usage instructions for testing on mobile.

8) Writeup & Submission (Reproducibility)

Paper (4–6 pages): problem, data, pipeline, augmentation, model, training, results, failures, ethics/safety, and limitations.
Submit to Canvas:

- PDF writeup (4–6 pages)
- Colab notebook with saved outputs
- Trained model weights

Your dataset section should include *some example images only* (no raw videos).

Submission Checklist

- **PDF** report (4–6 pages).
- **Colab notebook** with saved outputs.
- **Trained model weights**.
- **Failure-case gallery** (figure with captions).
- **Example dataset images** shown in the report.
- **Gradio app** usage instructions.

Grading Rubric (100 pts)

Criterion	**Pts**
Problem definition & engineering motivation	15
Data quality & splits (multi-session, no leakage)	20
Pipeline & augmentation (sound, reproducible)	20
Model & training (transfer learning done right)	25
Evaluation (accuracy + failure analysis)	10
Deployment (working Gradio app on mobile)	10
Documentation & repo hygiene	0
Bonus: domain-shift test; simple MLOps	+10

Constraints, Safety, and Ethics

- Avoid unsafe data collection (no hazards such as open flames or unattended appliances).
- Respect privacy; avoid faces and personal identifiers.
- Be explicit about model limits: this is a *support* tool, not a safety-certified device.

Deliverable Figures (Required)

1. Dataset overview: class counts per split (bar chart or table).
2. 6–8 failure cases with short captions (why the model failed).
3. Example images from your dataset.

Academic Integrity

Use only your own recordings and code except where libraries or starter snippets are explicitly allowed. Cite any external resources you consult. Collaboration is highly encouraged.