Project 1: Emotion-Change Detection for Classroom Feedback

Initial Proposal • October 27, 2025

> **To-Do:** Add the specific Kaggle dataset URL and citation once selected.

1) Motivation

(A) Problem. In typical learning environments, instructors lack accurate, real-time feedback about how their teaching methods affect students' emotions. It's nearly impossible to track class-wide emotional responses as they happen and do so reliably.

(B) Why this matters for AI/ML. Modern ML methods can quantify qualitative cues and reveal hidden patterns in emotional responses to instruction. Turning facial expressions into measurable signals enables evidence-based adjustments to teaching.

(C) Real-world motivation. Personalized teaching improves when educators can see which methods foster positive, sustained engagement. Better feedback loops can support equity by helping instructors adapt techniques to individual needs.

2) Task Definition

(A) Inputs → Outputs.
- **Input:** two facial images of the *same student* captured at different times (e.g., snapshots every ~10 s).
- **Output:** JSON with the change (delta) in confidence for each emotion: `anger`, `disgust`, `fear`, `happiness`, `neutrality`, `sadness`, `surprise`.

(B) Concrete I/O example (Python-style pseudocode).

```python
# Not executable; senior-engineer-readable pseudocode
def detect_emotion_change(image_path_1: str, image_path_2: str) -> dict:
    # 1) Load images from paths (same student, different times)
    # 2) Run both through a pre-trained facial emotion classifier
    #    -> returns per-emotion confidence scores for each image
    # 3) Compute deltas: scores_t2 - scores_t1 for each of 7 emotions
    # 4) Return JSON-like dict of deltas (no persistence, no PII)

    return {  # example shape
        "anger": d_anger,
        "disgust": d_disgust,
        "fear": d_fear,
        "happiness": d_happiness,
        "neutrality": d_neutrality,
        "sadness": d_sadness,
        "surprise": d_surprise
    }
```

(C) Task type. Multi-class **classification** (emotion recognition), summarized as per-class deltas between two snapshots.

3) Baseline

Use a well-tested, off-the-shelf facial emotion classifier to score each image separately, then compute per-emotion deltas (**t2 − t1**). Baseline ends at delta computation; integration with lecture timelines is **out of scope** for Project 1.

4) Proposed Approach

- Adopt a vetted foundation model for facial emotion recognition (e.g., transformer-based) with documented evaluations.
- Record why it is appropriate (reported accuracy, training data traits, published benchmarks); cite model card/paper.
- Implement a minimal inference wrapper that: (i) validates two input paths, (ii) runs emotion scoring, (iii) returns JSON deltas.
- *(Optional stretch)* Accept video input, auto-sample frames, and alert only on large deltas. (Not required for Project 1.)

5) Evaluation Plan

(A) Dataset. *TBD Kaggle dataset (facial emotion classification). Add URL/citation once finalized.*

(B) Metrics.
- **Macro-F1 (primary):** average F1 across the seven emotions (per-image classification).
- **Delta accuracy (secondary):** where paired ground-truth labels exist, report mean absolute error between predicted and true deltas per class.
- **Note:** ROC-AUC is less informative here due to the multi-class setup and is not a primary metric.

(C) Comparisons.
1) Report benchmark metrics from the model card on the chosen dataset.
2) External sanity check: a black-and-white facial-expression poster set (e.g., “Ernest/Varney” style), resized/normalized as a small out-of-distribution probe.

(D) Quantitative & qualitative notes.
- *Quantitative:* per-class confidence scores and macro-F1.
- *Qualitative limitations:* labeler bias; single-label simplification (mixed emotions not modeled); cultural/context variability in expressions.

6) Plan & Milestones (week-by-week sketch)

- **Week 1:** Select model and dataset; set up inference; reproduce minimal baseline on a small batch.
- **Week 2:** Implement delta computation and JSON output; add basic input validation/logging.
- **Week 3:** Run evaluations (macro-F1, delta checks); document limitations and error cases.
- **Week 4:** Polish write-up (figures/tables), package code repo, and finalize citations.

Appendix A: Brief Metrics Primer

- **Precision:** of predicted positives for a class, the fraction that are correct.
- **Recall:** of actual positives for a class, the fraction correctly identified.
- **F1:** harmonic mean of precision and recall; **macro-F1** averages F1 equally across classes.
- **Why not ROC-AUC?** Most natural for binary classification; multi-class ROC-AUC (one-vs-rest) is less informative here than macro-F1.
- **Delta accuracy:** summarize mean/median absolute error of predicted vs. true deltas across paired samples.