AUPP Robotics Lab – AI Robot Project Assignment Form

Project Title: Facial-Expression Follower – YOLOv8-face Detection + FER Classifier (Raspberry Pi 4)

Student Name(s): _____

Project Overview

Build a two-stage real-time system on Raspbot V2: (1) detect all faces using YOLOv8-face, (2) classify each detected face's emotion (FER) with a lightweight CNN (e.g., MobileNetV2 head). The robot selects a target face (e.g., most centered or user-selected track ID), infers emotion each frame, and executes an action (move/turn/stop) based on the predicted emotion. The design supports multi-person scenes, target lock, and re-acquisition.

Learning Objectives

- Integrate a multi-face detector (YOLOv8-face, ONNX on Raspberry Pi 4).
- Train or fine-tune a compact FER classifier (e.g., MobileNetV2 head) on FER2013/AffectNet subset.
- Implement a two-stage pipeline: detector → crops → classifier → control.
- Support multi-person selection, target lock, and re-acquisition logic.
- Deploy and evaluate performance (FPS, accuracy, and robot responsiveness).

System Architecture & Pipeline

1. Camera capture (640×480 or 424×240)
2. YOLOv8-face (ONNX, ~320–416 input) → face boxes + scores
3. Crop & resize each box (e.g., 224×224) → FER classifier (TFLite/ONNX)
4. Choose target face (center/size or locked track ID)
5. Map emotion → robot action; smooth decisions over time
6. Scan/reacquire if target lost N frames

Datasets & Preprocessing

- **Face Detection:** pre-trained YOLOv8-face weights (no training required initially).
- **FER Classification:** FER2013 (7 classes) or AffectNet subset (5–8 classes). Use 5 classes for control mapping: Happy, Sad, Angry, Surprised, Neutral.
- **Preprocess:** align/crop face, resize to 224×224 RGB, normalize to $[0,1]$; optional CLAHE for low light.

FER Classifier (MobileNetV2 Head)

- **Backbone:** MobileNetV2 (imagenet weights, include_top=False) + GAP + Dense(128, ReLU) + Dropout(0.3) + Dense(num_classes, Softmax)

- **Training:** 30–50 epochs, Adam (1e-4), early stopping on val_loss; augment (flip, brightness, slight rotation).
- **Export:** TensorFlow Lite (float16) or ONNX. Validate confusion matrix and per-class F1.

Raspberry Pi 4 Runtime & Optimization

- **Detector:** ONNX Runtime, imgsz 320–416; run every frame or every 2 frames.
- **Classifier:** TFLite/ONNX, batch the crops from all faces per frame if possible.
- **Tracking (optional):** SORT/ByteTrack for stable IDs and target lock.
- **Performance tips:** 64-bit OS, use smaller input, ROI around last target, EMA smoothing on probabilities.

Target Selection & Control

- **Selection:** choose the face with center closest to image center, or the largest box; allow user to lock a track ID.
- **Emotion decision:** majority vote over last 5 frames or EMA on class probabilities.
- **Control mapping** (example below) drives TB6612/L298N with deadband and max speed limits.

Emotion → Robot Action Mapping (Example)

Emotion	Robot Action
Happy	Move Forward
Angry	Move Backward
Surprised	Turn Left
Sad	Turn Right
Neutral	Stop/Idle

Safety & Robustness

- Gate on detector score ≥ 0.5 and FER max-prob ≥ 0.5.
- If no target for N frames → stop and slow-scan.
- Avoid rapid oscillation with hysteresis and low-pass filtering of commands.

Evaluation & Reporting

- **Detector performance:** FPS and detection rate under varied lighting.
- **FER accuracy:** overall, per-class F1, confusion matrix.
- **End-to-end:** reaction latency, stability, and success across 20 real-world trials.
- **Ablations:** single- vs multi-face, with/without tracking, window size for smoothing.

Deliverables

Type	Description	Format
FER Model	Trained MobileNetV2 FER (.h5) + TFLite/ONNX	Files
Runtime Code	YOLOv8-face ONNX + FER classifier + control	GitHub/ZIP
Demo Video	Robot reacting to emotions in multi-person scenes	MP4
Report	Metrics, latency, ablations, and lessons learned	PDF/Word
Slides	Overview, architecture, and demo	PPT/Google Slides