5) Dataset Usage Report（數據使用報告）

Date: 2025-12-24

Goal

Define which cleaned manifest is used for which training/evaluation stage, and why.

Key manifests

- Unified cleaned manifest (includes ExpW):
  - `Training_data_cleaned/classification_manifest.csv`
  - Validated in `outputs/manifest_validation_all_with_expw.json` (466,284 rows, 0 missing paths)

- HQ training manifest (used for most teacher/student training runs):
  - `Training_data_cleaned/classification_manifest_hq_train.csv`
  - CSV row count is **259,004** data rows (+ 1 header line)

- Mixed-source test manifest (deployment-realistic):
  - `Training_data_cleaned/test_all_sources.csv` (48,928 rows; used in ensemble selection experiments)

- Dataset-specific evaluation manifests:
  - `Training_data_cleaned/rafdb_basic_only.csv`
  - Other RAF/compound/RAF-ML evaluation CSVs under `Training_data_cleaned/`

Recommended usage (current project)

- Teacher Stage A training:
  - Use `classification_manifest_hq_train.csv` for strong, clean supervision.

- Softlabel export for KD/DKD:

  - Export teacher logits on the same training manifest used for student training.
  - Artifacts live in `outputs/softlabels/.../softlabels.npz` and `softlabels_index.jsonl`.

- Student CE/KD/DKD training:
  - Use the same `classification_manifest_hq_train.csv` to keep data distribution aligned.

- Ensemble selection / robustness evaluation:
  - Use `test_all_sources.csv` to pick weights that generalize to mixed sources.

Why multiple manifests

- `classification_manifest.csv` maximizes coverage (includes ExpW) and supports broader experiments.
- `classification_manifest_hq_train.csv` prioritizes higher-quality training signals.
- `test_all_sources.csv` is used for stress-testing generalization.

Consistency checks

- Canonical class list is enforced across datasets:
  - `Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral`
- Manifest integrity validation files:
  - `outputs/manifest_validation.json`
  - `outputs/manifest_validation_all_with_expw.json`

Next steps

- Add a short README table mapping each training script to its manifest argument, so runs are reproducible by inspection.