

Phase Plan

Đề tài. Chẩn đoán lỗi ổ lăn từ tín hiệu rung dựa trên kết hợp đặc trưng sâu và nông (CWRU làm bộ dữ liệu lỗi).

Bài báo. *Vibration signal analysis for rolling bearings faults diagnosis based on deep-shallow features fusion.*

Tài liệu môn học. Bishop — *Pattern Recognition and Machine Learning* (2006).

Phương pháp (trong đề án). Chuỗi xử lý triển khai theo nội dung slide: tín hiệu rung \rightarrow log-mel (96×64) \rightarrow (MBH-LPQ + CNN) \rightarrow so khớp cosine \rightarrow tổng trọng số điểm (WS) với quét α (WS sweep), đồng thời quét tham số b của MBH-LPQ (b sweep).

Ghi chú: bài báo gốc có thêm VGGish + PCA + EDA trước cosine; trong đề án em giữ cơ chế matching + fusion và thay nhánh sâu bằng CNN.

Mô hình. Nhánh đặc trưng nông (LBP/LDP/LPQ và biến thể MBH-LPQ) và nhánh sâu (CNN trên log-mel); đánh giá baseline ML, CNN, và fusion ở mức đặc trưng (feature-level) và mức điểm (score-level).

Nguyên tắc chung. Mỗi phase có đầu ra rõ ràng, tham số lưu bằng tệp cấu hình, seed cố định; tách bạch train/val/test; kết quả có thể tái lập.

Phase 0 — Xác lập đề tài và phạm vi

Mục tiêu:

Chốt tên đề tài, mục tiêu, phạm vi ứng dụng.

Nội dung: • Chuẩn hoá thuật ngữ và phạm vi đánh giá.

- Tài liệu hoá mục tiêu và kỳ vọng kết quả.

Kết quả:

Tệp mô tả phạm vi và mục tiêu nằm trong [docs/requirements.md](#).

Phase 1 — Khung dự án, môi trường và tái lập

Mục tiêu:

Tạo khung thư mục, môi trường conda, quy trình tái lập.

Nội dung: • Cấu trúc: configs/, data/, docs/, experiments/, notebooks/, original_paper/, results/, src/, tests/.

- `environment.yml` cho môi trường `ndm`; cố định seed trong cấu hình.
- Quản lý dữ liệu lớn bằng Git LFS (ví dụ `*.mat`) hoặc gói tải về qua Release (tùy thiết lập repo).
- README có mục hướng dẫn chạy lại và nơi đặt artefact (hình, bảng, confusion matrix).

Kết quả:

Dự án cài đặt và chạy được; [docs/reproducibility.md](#).

Phase 2 — Đọc bài báo và chốt “paper-core” theo đề án

Mục tiêu:

Nắm rõ pipeline bài báo và chốt phần “core” em tái hiện trong đề án.

Nội dung:

- Tóm tắt bài báo, trích các khối chính và mục tiêu của fusion.
- Chốt pipeline triển khai thực tế trong đề án (đúng với slide): `log-mel` \rightarrow (MBH-LPQ + CNN) \rightarrow cosine \rightarrow WS, kèm sweep b và α .
- Bảng ánh xạ khối xử lý \leftrightarrow nội dung môn học (PRML).

Kết quả:

`papers/sciRep2025_summary.md`, `docs/course_mapping.md`.

Phase 3 — Dữ liệu: tải, mô tả, hệ thống nhãn

Mục tiêu:

Tổ chức dữ liệu CWRU đầy đủ và nhất quán (load 0–3).

Nội dung:

- Tổ chức `data/raw/CWRU/`: nhóm `normal_baseline/` và `48k_drive_end_fault/`.
- Chuẩn hoá nhãn 10 lớp: Normal (1) + IR (3) + Ball (3) + OR (3), đủ các load 0–3.
- Sinh metadata: `data/labels.csv` và `data/splits/cwru_splits.json`.

Kết quả:

`docs/data_description.md`, `data/labels.csv`, `data/splits/cwru_splits.json`.

Phase 4 — Nạp dữ liệu, tách tập, cắt đoạn/chuẩn hoá

Mục tiêu:

Xây dựng mô-đun nạp dữ liệu tin cậy, không rò rỉ.

Nội dung:

- Tách `train/val/test` theo *file* (hoặc theo load) trước khi cắt đoạn để tránh leakage.
- Cắt đoạn thành mẫu huấn luyện; chuẩn hoá biên độ theo cấu hình.
- Tham số hoá qua `configs/*` và có kiểm tra nhanh bằng notebook.

Kết quả:

`src/dataio.py`, `notebooks/01_explore_data.ipynb`.

Phase 5 — Tuyển cơ sở ML theo PRML

Mục tiêu:

Thiết lập mốc so sánh cổ điển (baseline).

- Nội dung:**
- Trích đặc trưng cơ bản (thời gian/tần số hoặc thống kê) và huấn luyện Logistic/Softmax, SVM, Random Forest.
 - Chọn tham số trên validation, báo cáo trên test; xuất confusion matrix.

Kết quả:

`results/baseline_prml_metrics.csv`, `results/cm_prml.png`.

Phase 6 — Log-mel chuẩn hoá + audit hình ảnh

Mục tiêu:

Chuẩn hoá biến đổi thời-tần và đảm bảo log-mel đúng trước khi train.

- Nội dung:**
- STFT \rightarrow Mel \rightarrow Log; cố định cửa sổ, overlap, `nfft`, số dải mel.
 - Chuẩn kích thước log-mel 96×64 (phù hợp pipeline slide).
 - Tạo “audit log-mel” để kiểm tra trực quan (phổ bị bệt/lệch dải/mất cấu trúc).

Kết quả:

`src/spectrogram.py`, `results/*audit_logmel*.png`, `notebooks/04_spectrogram_preview.ipynb`.

Phase 7 — CNN cơ sở (nhánh sâu)

Mục tiêu:

Có mô hình sâu chạy được ổn định trên log-mel và xuất confusion matrix.

- Nội dung:**
- Huấn luyện CNN đa lớp với softmax + cross-entropy trên log-mel.
 - Lưu checkpoint và xuất embedding/điểm số nếu cần cho fusion.

Kết quả:

`results/cnn_baseline.pt`, `results/cm_cnn.png`, (tùy chọn) `X__deep.npy`.

Phase 8 — Fusion: feature-level và score-level

Mục tiêu:

Đánh giá kết hợp sâu-nông theo 2 kiểu: nối đặc trưng và WS theo điểm.

- Nội dung:**
- Feature-level: nối vector (deep + shallow) rồi huấn luyện bộ phân lớp.
 - Score-level: cosine matching tạo điểm số, sau đó WS fusion theo trọng số α .

Kết quả:

`results/cm_fusion_feature.png`, `results/cm_fusion_score.png`, và các bảng metric tương ứng.

Phase 9 — “Paper-core” theo đúng nội dung đề án (tái hiện + sweep)

Mục tiêu:

Tái hiện phần “core” đúng như slide: log-mel + MBH-LPQ + CNN + cosine + WS, có sweep.

Nội dung: • Dữ liệu lỗi: CWRU 10 lớp, kích bản theo load (0–3); split 80/20 và có validation (nếu dùng).

- Log-mel 96×64 (kèm audit).
- Nhánh nông: MBH-LPQ và (tùy chọn) LBP/LDP/LPQ để ablation shallow.
- Nhánh sâu: CNN trên log-mel (thay cho VGGish trong paper gốc).
- So khớp cosine và WS fusion: quét b (MBH-LPQ) và quét α (WS sweep), xuất biểu đồ sweep + confusion matrix.

Kết quả:

results/phase9/*cm_paper_core*.png,

results/phase9/*b_sweep_plot*.png, results/phase9/*ws_sweep_plot*.png, và file tổng kết cấu hình tốt nhất.

Phase 10 — Mở rộng: domain shift theo tải (load 0–3)

Mục tiêu:

Đánh giá khả năng khái quát khi thay đổi điều kiện tải.

Nội dung: • Thiết lập train load A, test load B với $A, B \in \{0, 1, 2, 3\}$.

- Tổng hợp heatmap domain shift và minh hoạ bằng confusion matrix cho một số cặp tiêu biểu (ví dụ $0 \rightarrow 0$ và $0 \rightarrow 3$).

Kết quả:

results/phase10/*cm_domain_shift*.png, results/phase10/domain_shift_summary.csv, heatmap tổng hợp.

Phase 11 — Ablation và kiểm thử nhiễu

Mục tiêu:

Hiểu vai trò các thành phần và độ bền mô hình.

Nội dung: • Ablation: baseline ML vs CNN vs shallow vs fusion; feature-level vs score-level; ảnh hưởng tham số log-mel.

- Kiểm thử nhiễu Gaussian theo SNR (nếu triển khai): $\{-6, -3, 0, 3, 6\}$ dB.

Kết quả:

results/ablation_plots.png, các bảng tổng hợp và ghi chú phân tích.

Phase 12 — Tự động hoá chạy và phát hành Release

Mục tiêu:

Một lệnh tái tạo toàn bộ và gom artefact.

Nội dung: • Script chạy tổng: `python experiments/run_all.py` (hoặc tương đương) theo cấu hình.

- Sinh manifest (commit, seed, phiên bản thư viện, danh sách artefact).
- Đóng gói Release: code + report + ảnh kết quả; dữ liệu có thể để LFS hoặc hướng dẫn tải riêng.

Kết quả:

Artefact đầy đủ trong `results/` và trang Release vX.Y.Z.

Phase 13 — Ghi chú kỹ thuật và công thức

Mục tiêu:

Chuẩn hoá nền tảng kỹ thuật dùng trong đồ án.

Nội dung: • STFT/mel/log-mel, LBP/LDP/LPQ/MBH-LPQ, cosine similarity, WS fusion, SNR; định nghĩa và ý nghĩa.

Kết quả:

`docs/math_notes.md` (hoặc tài liệu tương đương).

Phase 14 — Báo cáo và trình bày

Mục tiêu:

Hoàn thiện hồ sơ nộp và trình bày.

Nội dung: • Soạn report, slide.

Kết quả:

`Report.pdf`, `slides.pdf`, thẻ v1.0.