

- **Chủ đề chính:** Cuộc thi tập trung vào AI/ML cơ bản đến nâng cao (theo syllabus IOAI 2025), bao gồm lập trình Python, ML cổ điển, Neural Networks/Deep Learning, Computer Vision (CV), Natural Language Processing (NLP). Có phần thi trắc nghiệm (kiến thức lý thuyết/practice), lập trình thực tế (task như phân loại văn bản, dịch máy, khôi phục hình ảnh, semantic segmentation, matching icons, authentication paintings, radar processing).
- **Đặc thù cuộc thi:** Sử dụng Jupyter Notebook với thư viện sẵn (PyTorch, TensorFlow, Scikit-learn, Transformers, v.v.); cho phép dùng LLM (deepseek-r1-distill-qwen-32b) trong 5 giờ đầu để hỗ trợ tư duy/phát triển mô hình; thời gian thi 6 giờ; tập dữ liệu có training_set, public_test (mở luôn), private_test (mở giờ cuối).
- **Thời gian:** Chỉ còn khoảng 1 tuần, nên roadmap tập trung vào ôn nhanh, thực hành, ưu tiên chủ đề "Both" và "Practice" từ syllabus (vì thi thực tế nhiều). Mỗi ngày dành 4-6 giờ ôn, kết hợp lý thuyết + code.

Roadmap được chia thành 2 phần chính: (1) Ôn tập kiến thức AI/ML, (2) Prompting LLM. Mỗi phần có lịch cụ thể, nhiệm vụ hàng ngày, và tài liệu tham khảo (link hoặc sách/repo miễn phí, dễ tải).

Phần 1: Roadmap Ôn Tập Kiến Thức AI/ML

Dựa trên syllabus IOAI 2025 và các đề thi (trắc nghiệm sơ loại, task ngày 1-2 IOAI, đề VOAI chung kết). Ưu tiên:

- Thực hành code trên Jupyter (dùng thư viện sẵn như trong hướng dẫn thi).
- Ôn task thực tế: Phân loại/dịch văn bản (NLP), khôi phục hình ảnh (CV), semantic segmentation (Radar), matching icons (CV), authentication paintings (Semi-supervised ML).
- Sử dụng baseline.ipynb từ các task trong tài liệu để thực hành.

Ngày 1: Foundational Skills & Programming Fundamentals (Practice-focused)

- **Mục tiêu:** Nắm vững Python cơ bản, data handling, visualization – nền tảng cho tất cả task.
- **Nhiệm vụ:**
 - Ôn Python basics: Loops, functions, NumPy/Pandas (array manipulation, data loading CSV/JSON).
 - Thực hành: Load dữ liệu từ training_set.csv (như trong Antique Painting Authentication), visualize với Matplotlib/Seaborn.
 - Ôn Reproducibility: Set seed, CPU/GPU training (dùng PyTorch basics).
 - Thực hành task mẫu: Xử lý dữ liệu từ đề trắc nghiệm (Câu 6: Bag-of-Words với PyTorch).
- **Thời gian:** 2 giờ lý thuyết + 3 giờ code.
- **Tài liệu tham khảo:**
 - Khóa học: "Python for Data Science" trên Coursera (Andrew Ng) – Module 1-2 (miễn phí audit: <https://www.coursera.org/learn/python-for-applied-data-science-ai>).
 - Sách: "Python Crash Course" (Eric Matthes) – Chương 1-5 (PDF miễn phí: https://ehmatthes.github.io/pcc_2e/).
 - Repo: Hugging Face Datasets tutorial (<https://huggingface.co/docs/datasets/quickstart>) – Thực hành load data.

Ngày 2: Classical Machine Learning (Supervised & Unsupervised – Both/Practice)

- **Mục tiêu:** Ôn ML cổ điển, evaluation – Áp dụng cho task như Antique Painting Authentication (semi-supervised, labels unknown).
- **Nhiệm vụ:**
 - Supervised: Linear/Logistic Regression, KNN, Decision Trees, SVM, Random Forests, Gradient Boosting (XGBoost).
 - Unsupervised: K-Means, PCA, t-SNE, DBSCAN.
 - Evaluation: Metrics (Accuracy, F1, ROC), Overfitting/Underfitting, Cross-Validation, Hyperparameter Tuning.
 - Thực hành: Train model trên Scikit-learn cho phân loại (dùng training_set.csv từ task Painting), tune hyperparams với GridSearchCV.
 - Ôn câu trắc nghiệm: Câu 2 (Convolutional filters), Câu 3 (Loss không giảm).
- **Thời gian:** 2 giờ lý thuyết + 3 giờ code (thử semi-supervised với KMeans + labels known/unknown).
- **Tài liệu tham khảo:**
 - Sách: "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" (Aurélien Géron) – Chương 2-6 (PDF: <https://github.com/ageron/handson-ml3>).
 - Khóa học: "Machine Learning" (Andrew Ng on Coursera) – Week 1-6 (miễn phí: <https://www.coursera.org/learn/machine-learning>).
 - Repo: Scikit-learn examples (https://scikit-learn.org/stable/auto_examples/index.html) – Thực hành clustering/classification.

Ngày 3: Neural Networks & Deep Learning (Both/Practice)

- **Mục tiêu:** Nắm DL cơ bản – Áp dụng cho task Radar (semantic segmentation), Icon Matching (matching model).
- **Nhiệm vụ:**
 - Neural Nets: Perceptron, Gradient Descent, Backpropagation, Activation (ReLU/Sigmoid), Loss (MSE/Cross-Entropy).
 - DL: MLP, SGD/Mini-Batch, Optimizers (Adam), Regularization (Dropout, Batch Norm), Early Stopping.
 - Thực hành: Build MLP trên PyTorch cho classification; train trên GPU (dùng Weights & Biases cho tracking).
 - Ôn câu trắc nghiệm: Câu 1 (Batch Norm vị trí), Câu 3 (Loss không giảm).
- **Thời gian:** 2 giờ lý thuyết + 3 giờ code (thử baseline từ task Radar: Input 6 heatmaps → Output label map).
- **Tài liệu tham khảo:**
 - Khóa học: "Deep Learning Specialization" (Andrew Ng on Coursera) – Course 1-2 (miễn phí audit: <https://www.coursera.org/specializations/deep-learning>).
 - Sách: "Deep Learning" (Ian Goodfellow) – Chương 6-8 (PDF miễn phí: <https://www.deeplearningbook.org/>).
 - Repo: PyTorch tutorials (<https://pytorch.org/tutorials/>) – Beginner section.

Ngày 4: Computer Vision (Both/Practice)

- **Mục tiêu:** Ôn CV – Trọng tâm task Icon Matching, Painting Authentication, Khôi phục kiệt tác (Image Restoration từ đề VOAI).
- **Nhiệm vụ:**

- Fundamentals: Conv Layers, Pooling, Image Classification, Object Detection (YOLO basics), Segmentation (U-Net).
- Advanced: Transfer Learning (ResNet/MobileNet), Augmentation, Feature Extraction, ViT, GANs.
- Thực hành: Dùng torchvision load images; train matching model (crop/orig icons); thử inpainting cho restoration.
- Ôn task: Mask rectangle cho images (từ task Day 2), semantic segmentation trên heatmaps.
- ORB / SIFT feature matching (OpenCV).
- SSIM (structural similarity) để đánh giá cặp ảnh.
- Xếp lại mảnh bằng heuristic (góc trái trên → greedy matching).
- Transfer learning: ResNet / MobileNet → nhanh, nhẹ.
- **Thời gian:** 2 giờ lý thuyết + 3 giờ code (sử dụng OpenCV/Pillow cho processing).
- **Tài liệu tham khảo:**
 - Khóa học: "Convolutional Neural Networks" (Andrew Ng on Coursera) – Part of DL Specialization.
 - Sách: Géron's book – Chương 14 (CV with TensorFlow).
 - Repo: Hugging Face Transformers for CV
(https://huggingface.co/docs/transformers/tasks/image_classification) – Thực hành transfer learning.
 - [OpenCV Feature Matching](#)
 - [MobileNet for Transfer Learning](#)
 - [SSIM — scikit-image docs](#)

Ngày 5: Natural Language Processing (Practice-focused)

- **Mục tiêu:** Ôn NLP – Áp dụng cho task Phân loại/Dịch văn bản (từ đề VOAI).
- **Nhiệm vụ:**
 - Word Embeddings (Word2Vec/GloVe), Transformers (Attention), Text Classification, Pre-trained (BERT/GPT).
 - Fine-tuning (LoRA), LLM Agents, Chatbots.
 - Thực hành: Fine-tune BERT cho classification/translation trên Hugging Face; dùng datasets library.
 - Ôn câu trắc nghiệm: Câu 4 (Fine-tune GPT), Câu 6 (BoW với PyTorch).
- **Thời gian:** 2 giờ lý thuyết + 3 giờ code (thử task VOAI: Classify/Translate text).
- **Tài liệu tham khảo:**
 - Khóa học: "Natural Language Processing Specialization" (DeepLearning.AI on Coursera) – Course 1-2.
 - Sách: "Speech and Language Processing" (Jurafsky) – Chương 6-9 (PDF: <https://web.stanford.edu/~jurafsky/slp3/>).
 - Repo: Hugging Face NLP tutorials
(https://huggingface.co/docs/transformers/tasks/sequence_classification).

Ngày 6: Thực Hành Đề Thi & Task Mẫu

- **Mục tiêu:** Áp dụng toàn bộ kiến thức vào đề thực tế.
- **Nhiệm vụ:**
 - Giải đề trắc nghiệm sơ loại (De-thi-trac-nghiem-soloai-public.pdf) – Kiểm tra đáp án tự suy luận.
 - Thực hành task từ IOAI Day 1-2: Radar (segmentation), Icon Matching, Painting (semi-supervised), Masking images.

- Thực hành đề VOA1: Task 1 (Classify/Translate), Task 2 (Image Restoration).
- Kiểm tra môi trường: Code trên Jupyter, không pip install, dùng thư viện sẵn.
- **Thời gian:** 5 giờ code + debug.
- **Tài liệu tham khảo:**
 - Baseline code từ tài liệu (baseline.ipynb cho mỗi task).
 - Kaggle notebooks tương tự: Search "semantic segmentation PyTorch" (<https://www.kaggle.com/search?q=semantic+segmentation>).

Ngày 7: Tinh chỉnh & tăng tốc

- **Nhiệm vụ:**
 - Giảm thời gian train/inference, tăng độ ổn định.
 - Batch size, augmentation, early stopping.
 - Sử dụng `torch.cuda.amp` (mixed precision) để tăng tốc.
 - Giảm thời gian EDA bằng caching pre-tokenized data.
- **Tài liệu:**
 - [PyTorch AMP](#)
 - [Mixed Precision Training Guide](#)

Phần 2: Roadmap Prompting LLM

Vì cuộc thi cho phép dùng LLM (deepseek-r1-distill-qwen-32b) trong 5 giờ đầu để hỗ trợ code/tư duy (không dùng cho inference cuối), cần học cách prompt hiệu quả để generate code, debug, hoặc ý tưởng model.

Ngày 1-2: Kiến Thức Cơ Bản Prompting (Tích Hợp Với Ôn ML)

- **Mục tiêu:** Hiểu zero-shot, few-shot.
- **Nhiệm vụ:** Thử prompt đơn giản: "Giải thích Batch Normalization và code ví dụ PyTorch."
- **Tài liệu:** OpenAI Prompt Engineering Guide (<https://platform.openai.com/docs/guides/prompt-engineering>) – Đọc Section 1-3.

Ngày 3-4: Kỹ Thuật Nâng Cao

- **Mục tiêu:** Chain-of-Thought (CoT), Role-playing.
- **Nhiệm vụ:** Prompt cho task: "Là expert ML, phân tích task Radar segmentation và suggest model architecture với PyTorch."
- **Tài liệu:** Anthropic's Prompting Guide (<https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering>) – Focus on CoT.

Ngày 5-6: Thực Hành Với Task Cuộc Thi

- **Nhiệm vụ:** Sử dụng prompt để generate code cho task VOA1 (e.g., "Fine-tune BERT cho text classification, code đầy đủ.").
- **Tài liệu:** Khóa học "Prompt Engineering for Developers" (DeepLearning.AI on Coursera) – Short course miễn phí (<https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/>).

Ngày 7: Tối Ưu Hóa

- **Nhiệm vụ:** Test prompt nhanh, tránh dài dòng; retry nếu output sai.
- **Tài liệu:** Paper "Chain-of-Thought Prompting" (Google) – PDF: <https://arxiv.org/pdf/2201.11903.pdf>.
- [Prompting for LLM Code Generation](#)

Lời khuyên chung: Thực hành trên Google Colab hoặc local Jupyter để quen môi trường. Theo dõi chia sẻ kinh nghiệm từ file PDF (Trần Quốc Long) về IOAI – Tập trung cá nhân lập trình. Nếu cần, join cộng đồng AI Việt Nam trên Facebook để hỏi thêm. Chúc thi tốt!