

OLYMPIC TRÍ TUỆ NHÂN TẠO 2025

TOPIC 3: COMPUTER VISION (CV) - KAGGLE VERSION

Phân vùng tổn thương phổi từ ảnh X-quang ngực

(Medical Image Segmentation with Explainable AI - Kaggle Platform)

Contents

1	HƯỚNG DẪN CHUNG	3
1.1	Thể thức thi đấu	3
1.1.1	Hướng dẫn sử dụng Kaggle Environment	3
1.2	Thư viện hỗ trợ	3
1.3	Quy định về dữ liệu và mã nguồn	3
1.4	Quy định về mô hình ngôn ngữ lớn (LLM)	4
1.5	Nộp bài và đánh giá	4
1.6	Bài tập bổ sung	4
2	MÔ TẢ BÀI TOÁN	4
2.1	Tổng quan	4
2.2	Mục tiêu học tập	4
3	MÔ TẢ DỮ LIỆU	5
3.1	Cấu trúc dữ liệu Kaggle	5
3.2	Hướng dẫn sử dụng Dataset Kaggle	5
3.2.1	Dataset chính thức	5
3.2.2	Cấu hình Kaggle Environment	6
3.2.3	Ưu điểm của Kaggle Platform	6
3.3	Thống kê dữ liệu	7
4	ĐÁNH GIÁ	7
4.1	Metrics chính	7
4.1.1	Dice Coefficient	7
4.1.2	IoU (Intersection over Union)	7
4.1.3	Binary F1-score	8
4.2	Yêu cầu visualization	8
4.3	Kết quả mong đợi	8
5	MÔ HÌNH BASELINE	8
5.1	Kiến trúc đề xuất - Simple U-Net	8
5.2	Cấu trúc đề xuất	9
5.3	Hướng dẫn bắt đầu Kaggle	10
5.4	Cấu hình Kaggle Optimization	10
6	YÊU CẦU NỘP BÀI	10
6.1	Deliverables bắt buộc	10
6.2	Criteria đánh giá	11
6.3	Deadline	11

7	GỢI Ý NÂNG CAO	11
7.1	Self-supervised Segmentation	11
7.2	Transformer-based models	11
7.3	Multimodal fusion	11
7.4	Advanced XAI	11
7.5	Ensemble methods	11
7.6	Kaggle Development Tips	11

1 HƯỚNG DẪN CHUNG

1.1 Thể thức thi đấu

- **Ngôn ngữ lập trình:** Python 3.10
- **Môi trường:** Kaggle Notebooks với GPU T4 x2
- **Thời gian:** 4 giờ
- **Hình thức:** Lập trình thực hành trên Kaggle platform

1.1.1 Hướng dẫn sử dụng Kaggle Environment

- **Tạo notebook:** Tạo notebook mới trên Kaggle.com
- **Bật GPU:** Settings → Accelerator → GPU T4 x2
- **Add dataset:** Click "Add Data" → Tìm "chest-xray-masks-and-labels-processed" → Add
- **Chạy code:** Chạy các cell theo thứ tự từ trên xuống dưới
- **Lưu kết quả:** Files sẽ được lưu trong thư mục /kaggle/working/

1.2 Thư viện hỗ trợ

Các thư viện sẽ được cài đặt tự động trong notebook:

- torch (PyTorch) với CUDA support
- torchvision
- numpy
- matplotlib
- scikit-learn
- Pillow
- opencv-python
- tqdm
- segmentation-models-pytorch
- albumentations
- captum

1.3 Quy định về dữ liệu và mã nguồn

- Thí sinh được phép sử dụng các thư viện đã cài đặt
- Không được sử dụng API ngoài hoặc dịch vụ cloud computing khác
- Tất cả mô hình phải chạy trên Kaggle platform
- Dataset đã được cung cấp sẵn và xử lý trong Kaggle input
- **Kaggle:** Dataset có sẵn tại /kaggle/input/chest-xray-masks-and-labels-processed/

- **Cấu trúc:** Train (226 ảnh) và Test (57 ảnh) đã được chia sẵn

1.4 Quy định về mô hình ngôn ngữ lớn (LLM)

- Không được sử dụng ChatGPT, Claude, Gemini hoặc các LLM khác trong quá trình thi
- Không được tra cứu code từ GitHub, Stack Overflow hoặc các nguồn khác
- Chỉ được sử dụng tài liệu tham khảo đã được cung cấp

1.5 Nộp bài và đánh giá

- Nộp file notebook (.ipynb) đã hoàn thành
- Kèm theo file PDF báo cáo kết quả (tối đa 5 trang)
- Đánh giá dựa trên: Độ chính xác mô hình (40%), Code quality (30%), Báo cáo (30%)
- Thời gian nộp bài: Trong vòng 30 phút sau khi kết thúc thi

1.6 Bài tập bổ sung

Ngoài bài chính, thí sinh có thể làm thêm các bài tập nâng cao để cộng điểm:

- Implement thêm các kỹ thuật XAI khác (Integrated Gradients, SHAP)
- Thử nghiệm với các kiến trúc mô hình khác (UNet++, DeepLabv3)
- Áp dụng data augmentation nâng cao
- So sánh hiệu suất với các pretrained encoders khác nhau

2 MÔ TẢ BÀI TOÁN

2.1 Tổng quan

Trong lĩnh vực chẩn đoán hình ảnh y học, việc phân vùng phổi trên ảnh X-quang ngực là một bài toán quan trọng giúp hỗ trợ bác sĩ trong việc phát hiện và chẩn đoán các bệnh lý về phổi. Nhiệm vụ của bạn là:

1. **Huấn luyện mô hình Simple U-Net** để phân vùng phổi trên ảnh X-quang ngực
2. **Sử dụng kỹ thuật XAI (Explainable AI)** để minh họa vùng mà model tập trung chú ý
3. **Đánh giá hiệu suất** bằng các metrics phù hợp cho bài toán segmentation

2.2 Mục tiêu học tập

- Hiểu và thực hành bài toán **Medical Image Segmentation**
- Làm quen với kiến trúc **Simple U-Net** và các biến thể
- Áp dụng **Explainable AI** (GradCAM) để giải thích model

- Sử dụng các **metrics phù hợp** cho segmentation (Dice, IoU, F1-score)
- Thực hành **Data Augmentation** và **Transfer Learning**

3 MÔ TẢ DỮ LIỆU

3.1 Cấu trúc dữ liệu Kaggle

Dataset "Chest X-ray Masks and Labels" đã được xử lý và tối ưu hóa với cấu trúc:

```
/kaggle/input/chest-xray-masks-and-labels-processed/
Lung Segmentation/
  train/                    # 226 ảnh (80%)
    CXR_png/                # Ảnh X-quang gốc
      CHNCXR_0001_0.png
    ...
    masks/                  # Mask phân vùng
      CHNCXR_0001_0_mask.png
    ...
    ClinicalReadings/       # Thông tin lâm sàng
  test/                     # 57 ảnh (20%) - Ấn đi
    CXR_png/
    masks/
    ClinicalReadings/
```

- **Tổng số ảnh gốc:** 800 ảnh X-quang ngực
- **Số ảnh có mask hợp lệ:** 566 ảnh (sau khi loại bỏ ảnh không có mask)
- **Số ảnh sử dụng:** 283 ảnh (50% reduction để tăng tốc training)
- **Kích thước dataset:** 2.4GB (đã được xử lý và tối ưu hóa)
- **Train set:** 226 ảnh (80%) - Chia thành 203 train / 23 validation (90/10)
- **Test set:** 57 ảnh (20%) - Ấn đi, không sử dụng trong training
- **Kích thước ảnh:** 256×256 pixels (resized từ 3000×2919)
- **Class balance:** 50/50 normal/tuberculosis

3.2 Hướng dẫn sử dụng Dataset Kaggle

3.2.1 Dataset chính thức

- **Tên:** Chest X-ray Masks and Labels (Processed Version)
- **Tác giả:** Nikhil Pandey (Original), Olympic AI Team (Processed)
- **Link gốc:** <https://www.kaggle.com/datasets/nikhilpandey360/chest-xray-masks-and-labels>
- **Kích thước:** 2.4GB (đã được xử lý và tối ưu hóa)

- **Số lượng ảnh:** 283 ảnh X-quang ngực (từ 800 ảnh gốc)
- **Phân chia dataset:** Train (226 ảnh), Validation (23 ảnh), Test (57 ảnh - ẩn đi)
- **Kích thước ảnh:** 256×256 pixels (resized từ 3000×2919)
- **Định dạng:** PNG (ảnh RGB + mask grayscale)
- **Class balance:** 50/50 normal/tuberculosis

3.2.2 Cấu hình Kaggle Environment

Bước 1: Tạo notebook mới trên Kaggle

1. Vào `kaggle.com` → Sign in
2. Click "Create Notebook" → "New Notebook"
3. Bật GPU: Settings → Accelerator → GPU T4 x2

Bước 2: Add dataset vào notebook

1. Click "Add Data" ở góc phải
2. Tìm kiếm "chest-xray-masks-and-labels-processed"
3. Click "Add" để tích hợp dataset
4. Dataset sẽ xuất hiện trong `/kaggle/input/`

Bước 3: Thiết lập đường dẫn trong code

```
# Đường dẫn dataset
DATA_DIR = "/kaggle/input/chest-xray-masks-and-labels-processed"
LUNG_SEG_DIR = os.path.join(DATA_DIR, "Lung Segmentation")
TRAIN_DIR = os.path.join(LUNG_SEG_DIR, "train")
```

Bước 4: Tạo thư mục output

```
# Tạo thư mục output
os.makedirs('/kaggle/working/models', exist_ok=True)
os.makedirs('/kaggle/working/plots', exist_ok=True)
os.makedirs('/kaggle/working/gradcam', exist_ok=True)
```

3.2.3 Ưu điểm của Kaggle Platform

- **GPU miễn phí:** GPU T4 x2 với hiệu suất cao
- **Không cài đặt:** Tất cả thư viện đã có sẵn
- **Dataset sẵn có:** Không cần tải về, tự động mount
- **Collaboration:** Dễ dàng chia sẻ và collaboration
- **Version control:** Tự động lưu version của notebook

Thông số	Giá trị
Tổng số ảnh gốc	800
Số ảnh có mask hợp lệ	566
Số ảnh sử dụng	283
Kích thước dataset	2.4GB
Kích thước ảnh	256×256 pixels
Định dạng ảnh	PNG (RGB)
Định dạng mask	PNG (Grayscale)
Train set	226 ảnh (80%)
Train/Validation split	203/23 (90/10)
Test set	57 ảnh (20%)
Class balance	50/50 normal/tuberculosis

Table 1: Thống kê dataset đã xử lý

3.3 Thống kê dữ liệu

4 ĐÁNH GIÁ

4.1 Metrics chính

Bài toán segmentation sẽ được đánh giá bằng các metrics sau:

4.1.1 Dice Coefficient

Dice coefficient đo độ tương đồng giữa prediction và ground truth:

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \quad (1)$$

Trong đó:

- A : Predicted mask
- B : Ground truth mask
- TP: True Positive
- FP: False Positive
- FN: False Negative

4.1.2 IoU (Intersection over Union)

IoU đo tỷ lệ giao nhau giữa prediction và ground truth:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \quad (2)$$

4.1.3 Binary F1-score

F1-score cân bằng giữa Precision và Recall:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

Với:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

4.2 Yêu cầu visualization

CÁC THÍ SINH PHẢI THỰC HIỆN:

- Hiển thị ảnh gốc, ground truth mask, predicted mask, và overlay
- Tạo heatmap XAI (GradCAM) để minh họa vùng chú ý của model
- Lưu 5 kết quả visualization tốt nhất vào /kaggle/working/predictions/
- So sánh prediction với ground truth trên validation set
- **BẮT BUỘC:** Đạt Dice Score > 0.85 trên tập validation
- **BẮT BUỘC:** Implement GradCAM để giải thích model

4.3 Kết quả mong đợi

Hình 1 minh họa kết quả mong đợi từ mô hình segmentation với GradCAM:

Giải thích hình:

- **Cột 1:** Ảnh X-quang gốc với Dice scores cao (0.934, 0.965, 0.974)
- **Cột 2:** Ground truth mask (màu trắng = vùng phổi)
- **Cột 3:** Predicted mask từ mô hình
- **Cột 4:** GradCAM heatmap (màu sáng = vùng quan trọng)
- **Cột 5:** Overlay GradCAM trên ảnh gốc để hiểu vùng model tập trung

5 MÔ HÌNH BASELINE

5.1 Kiến trúc đề xuất - Simple U-Net

CẤU HÌNH BẮT BUỘC CHO THÍ SINH:

- **Simple U-Net** với encoder từ đầu (không pretrained)
- **Encoder:** 4 blocks Conv2D + MaxPool (64, 128, 256, 512 filters)
- **Bottleneck:** 1024 filters

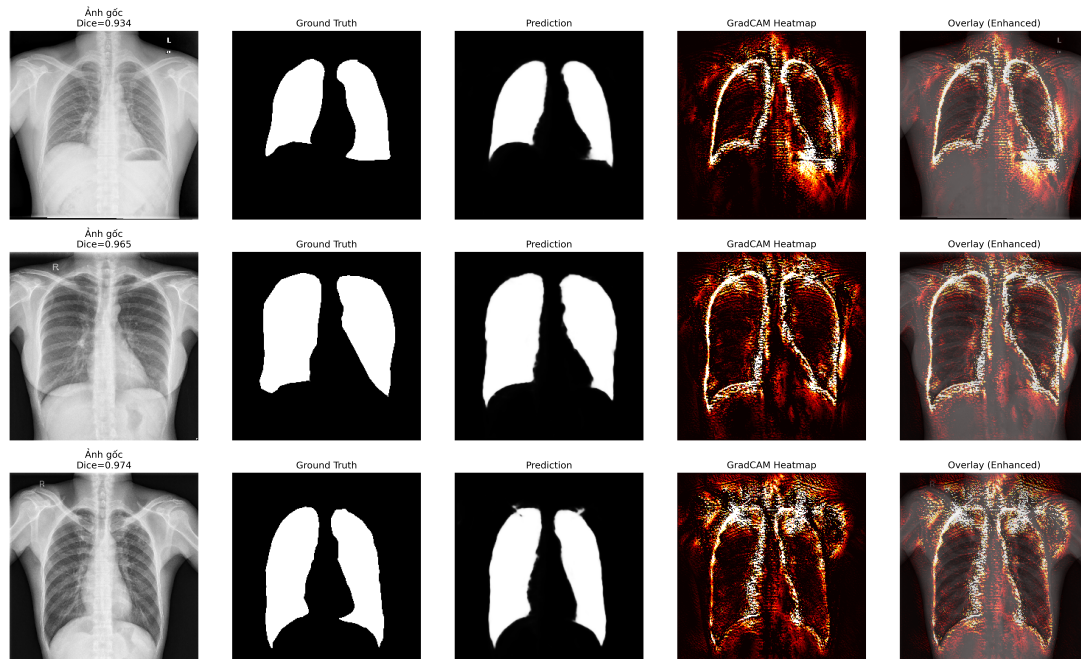


Figure 1: Kết quả mong đợi: Segmentation với GradCAM visualization. Các cột từ trái sang phải: Ảnh gốc, Ground Truth, Prediction, GradCAM Heatmap, Overlay (Enhanced). Dice scores cao (0.934, 0.965, 0.974) cho thấy chất lượng segmentation tốt.

- **Decoder:** ConvTranspose2D + skip connections
- **Loss function:** Combined Loss (Dice Loss + BCE Loss)
- **Optimizer:** Adam với learning rate 0.001
- **Batch size:** 4 (phù hợp với GPU T4)
- **Epochs:** 10 (quick experiment) hoặc 50+ (full training)
- **Image size:** 256×256 pixels
- **Data augmentation:** Horizontal flip, rotation, brightness/contrast adjustment
- **XAI:** GradCAM với target layer `model.enc4[-2]`
- **Workers:** 0 (tránh lỗi multiprocessing trên Kaggle)

5.2 Cấu trúc đề xuất

CÁC BƯỚC THÍ SINH PHẢI THỰC HIỆN:

1. **Data Loading & Preprocessing** (resize, augment, Kaggle paths)
2. **Model Initialization** (Simple U-Net architecture)
3. **Training & Evaluation** (train/val split với Kaggle config)
4. **Metric Calculation** (Dice, IoU, F1)
5. **XAI Implementation** (GradCAM heatmap generation)
6. **Visualization & Saving** (5 best results vào /kaggle/working/)

5.3 Hướng dẫn bắt đầu Kaggle

CÁC BƯỚC THÍ SINH PHẢI LÀM:

1. **Tạo notebook** mới trên Kaggle.com
2. **Bật GPU T4 x2** trong Settings
3. **Add dataset** "chest-xray-masks-and-labels-processed"
4. **Thiết lập DataLoader** với train/val split (203/23)
5. **Xây dựng Simple U-Net** với architecture từ đầu
6. **Huấn luyện model** với monitoring metrics
7. **Implement GradCAM** để tạo heatmap
8. **Visualize kết quả** và lưu model vào /kaggle/working/

5.4 Cấu hình Kaggle Optimization

CẤU HÌNH KHUYẾN NGHỊ:

- **Batch size:** 4 (phù hợp với GPU T4)
- **Epochs:** 10 (quick experiment) hoặc 50+ (full training)
- **Image size:** 256×256 (cân bằng giữa chất lượng và tốc độ)
- **Workers:** 0 (tránh lỗi multiprocessing trên Kaggle)
- **Device:** Tự động detect CUDA trên Kaggle
- **Learning rate:** 0.001 (Adam optimizer)
- **Scheduler:** ReduceLROnPlateau

6 YÊU CẦU NỘP BÀI

6.1 Deliverables bắt buộc

CÁC THÍ SINH PHẢI NỘP:

1. **Notebook hoàn chỉnh** (.ipynb) với tất cả cells đã chạy
2. **Model checkpoint** (.pth) đã được lưu trong /kaggle/working/models/
3. **Visualization results** (5 samples tốt nhất) trong /kaggle/working/predictions/
4. **GradCAM heatmaps** trong /kaggle/working/gradcam/
5. **Training plots** (loss, dice, IoU, F1) trong /kaggle/working/plots/
6. **Báo cáo PDF** (tối đa 5 trang) mô tả:
 - Approach và architecture
 - Kết quả metrics (Dice, IoU, F1)
 - Phân tích GradCAM results
 - Discussion và limitations

6.2 Criteria đánh giá

- **Model Performance** (40%): Dice Score > 0.85 trên validation set
- **Code Quality** (30%): Clean code, proper documentation, error handling
- **XAI Implementation** (20%): GradCAM visualization chất lượng cao
- **Report Quality** (10%): Báo cáo rõ ràng, phân tích sâu sắc

6.3 Deadline

- **Thời gian thi:** 4 giờ
- **Nộp bài:** Trong vòng 30 phút sau khi kết thúc thi
- **Hình thức nộp:** Upload lên hệ thống thi đấu

7 GỢI Ý NÂNG CAO

7.1 Self-supervised Segmentation

Sử dụng contrastive learning để pretrain encoder trên medical images trước khi fine-tune cho segmentation task.

7.2 Transformer-based models

Thử nghiệm với SegFormer, Vision Transformer, hoặc Swin Transformer cho segmentation.

7.3 Multimodal fusion

Kết hợp với thông tin lâm sàng (clinical notes) để cải thiện hiệu suất segmentation.

7.4 Advanced XAI

Implement thêm LayerGradCam, Integrated Gradients, SHAP để có nhiều góc nhìn về model interpretability.

7.5 Ensemble methods

Kết hợp nhiều model để cải thiện performance và robustness.

7.6 Kaggle Development Tips

- **Memory Management:** Sử dụng `torch.cuda.empty_cache()` khi cần
- **Checkpoint:** Lưu model định kỳ để tránh mất dữ liệu

- **Logging:** Sử dụng TensorBoard hoặc Weights & Biases
- **Profiling:** Sử dụng `torch.profiler` để tối ưu performance
- **GPU Utilization:** Monitor GPU usage để tối ưu batch size