

PHÂN ĐOẠN CỎ DẠI VÀ CÂY TRỒNG

Tên đề tài (Tiếng Việt): Nghiên cứu mô hình Phân đoạn Ngữ nghĩa Thời gian thực (Real-time Semantic Segmentation) thích ứng miền (Domain Adaptation) cho Robot nông nghiệp định vị và phun thuốc diệt cỏ chính xác.

Tên đề tài (Tiếng Anh - Suggested): *Precision Weed Management using Real-time Semantic Segmentation: A Lightweight Transformer-based Approach with Unsupervised Domain Adaptation for Agricultural Robots.*

1. Đặt vấn đề (Problem Statement)

- **Thực trạng:** Phun thuốc tràn lan (Broadcast spraying) gây lãng phí thuốc, ô nhiễm đất và tạo ra hiện tượng “cỏ kháng thuốc”.
- **Thách thức “Green-on-Green”:** Trước đây, việc phân biệt “Cây màu xanh trên nền Đất màu nâu” (Green-on-Brown) khá dễ. Nhưng hiện nay, thách thức là “**Cây màu xanh trên nền Cỏ màu xanh**” (**Green-on-Green**). Cỏ và cây trồng có màu sắc gần nhau y hệt nhau, chỉ khác nhau chút ít về hình dáng lá và kết cấu (texture).
- **Vấn đề môi trường không cấu trúc:** Robot di chuyển ngoài đồng sẽ gặp ánh sáng thay đổi liên tục (nắng/mây), bóng đổ, và đất đá gồ ghề làm camera bị rung.
- **Câu hỏi nghiên cứu:** Làm thế nào để xây dựng một mạng nơ-ron phân đoạn đạt tốc độ > 30 FPS trên thiết bị nhúng (Jetson/Raspberry Pi) để điều khiển voi phun thời gian thực, đồng thời giải quyết vấn đề “trượt miền” (Domain Shift) khi mang Robot từ ruộng này sang ruộng khác?

2. Mục tiêu nghiên cứu (Research Objectives)

1. **Phát triển kiến trúc lai CNN-Transformer hạng nhẹ:** Kết hợp khả năng trích xuất chi tiết cục bộ của CNN với khả năng hiểu ngữ cảnh toàn cục (Global Context) của Transformer (như SegFormer) để phân biệt cây và cỏ dại có hình dáng tương tự.
2. **Tối ưu hóa độ trễ (Inference Latency):** Cắt giảm mô hình để chạy trực tiếp trên biên (Edge Computing) với độ trễ < 50 ms (thời gian từ lúc camera nhìn thấy đến lúc voi phun kích hoạt).
3. **Thích ứng miền (Domain Adaptation):** Đề xuất giải pháp để mô hình huấn luyện trên dữ liệu ruộng ngô vẫn hoạt động tốt trên ruộng đậu tương hoặc trong điều kiện ánh sáng khác biệt mà không cần gán nhãn lại dữ liệu.

3. Tổng quan tài liệu & Khoảng trống nghiên cứu (Literature Review)

3.1. Các chỉ số thực vật truyền thống (Vegetation Indices):

- Trước kỷ nguyên AI, người ta dùng **ExG (Excess Green)** hoặc **NDVI** (nếu có camera quang phổ) để tách cây khỏi đất. Phương pháp này hoàn toàn thất bại trong bài toán Green-on-Green [1].

3.2. Mạng nơ-ron tích chập (CNNs) cho phân đoạn:

- U-Net:** Kiến trúc Encoder-Decoder kinh điển. Rất chính xác nhưng quá chậm và nặng nề cho Robot nhỏ.
- DeepLabV3+:** SOTA một thời, sử dụng Atrous Convolution để mở rộng vùng nhìn. Tuy nhiên, mô hình này yêu cầu phần cứng mạnh.

3.3. Vision Transformers (ViT) & SOTA hiện tại:

- SegFormer (NeurIPS 2021):** Loại bỏ Positional Encoding phức tạp, thiết kế nhẹ và hiệu quả. SegFormer đang chứng minh khả năng vượt trội so với CNN trong việc phân biệt các đối tượng có kết cấu phức tạp như cỏ dại [2].

3.4. Khoảng trống nghiên cứu (The Gap):

- Dữ liệu khan hiếm:** Hầu hết dataset hiện tại (như CWFID) chỉ chụp từ trên xuống (top-down) tĩnh. Thiếu dữ liệu video từ góc nhìn của Robot đang di chuyển (Motion blur included).
- Thiếu mô hình “Dynamic Adaptation”:** Các mô hình hiện tại thường “học vẹt”. Nếu đất thay đổi màu (từ đất đỏ sang đất đen), mô hình phân đoạn sai bét. Chưa có cơ chế tự điều chỉnh (Self-calibration) cho Robot.

4. Tài liệu tham khảo minh chứng (References)

Các công bố khoa học mới nhất (2022-2024) để đảm bảo tính thời sự:

- Về SegFormer trong Nông nghiệp:**
 - Paper:* “Real-Time Weed-Crop Segmentation in Smart Farming using Lightweight SegFormer” (Computers and Electronics in Agriculture, 2023).
 - Paper:* “Green-on-Green Weed Detection using Transformer-based Semantic Segmentation” (IEEE Access, 2024).
- Về Thích ứng miền (Domain Adaptation):**
 - Paper:* “Unsupervised Domain Adaptation for Weed Segmentation across Different Growth Stages” (Biosystems Engineering, 2023). Bài này giải quyết vấn đề cây lớn lên thì hình dáng thay đổi.
- Về Triển khai trên Robot (Robotics):**
 - Paper:* “Edge Computing for Real-Time Selective Spraying: A Comparative Study” (AgriEngineering, 2022).

5. Tài nguyên bước đầu (Resources Part 1)

A. Datasets (Dữ liệu chuẩn):

1. **Sugar Beets 2016:** Bộ dữ liệu robot nông nghiệp nổi tiếng nhất. Chứa ảnh RGB-D, định vị GPS, và nhãn phân đoạn chi tiết Cây/Cỏ/Đất. Dữ liệu này rất khó vì cây cỏ cải đường và cỏ dại rất giống nhau.
2. **CWFID (Crop/Weed Field Image Dataset):** Bộ dữ liệu nhỏ gọn, tốt để test thuật toán ban đầu.
3. **Agriculture-Vision (CVPR Challenge):** Dữ liệu chụp từ trên cao (Aerial view), số lượng cực lớn.

B. GitHub Repositories (Frameworks):

1. **MMSegmentation (OpenMMLab):** github.com/open-mmlab/mmsegmentation
 - Thư viện Phân đoạn số 1 thế giới hiện nay. Hỗ trợ sẵn **SegFormer**, **U-Net**, **PSPNet**. Sinh viên nên dùng thư viện này thay vì code PyTorch thuần để tiết kiệm 6 tháng công sức.
2. **Weed-Segmentation-SegFormer:** Tìm các repo cụ thể áp dụng SegFormer cho nông nghiệp để tham khảo config.

5. Phương pháp nghiên cứu & Kiến trúc đề xuất (Methodology)

Tôi đề xuất kiến trúc hệ thống “**Vision-to-Actuation Pipeline**” (Quy trình từ Thị giác đến Hành động) tích hợp trên nền tảng **ROS (Robot Operating System)**.

Module 1: Mô hình Phân đoạn Nhẹ (Lightweight Segmentation Model)

- **Backbone:** Sử dụng **SegFormer-B0** (phiên bản nhỏ nhất của dòng SegFormer).
 - *Tại sao chọn Transformer?* Khác với CNN (nhìn cục bộ), Transformer có cơ chế *Self-Attention* giúp nó so sánh kết cấu của cái lá ở góc trái với cái lá ở góc phải để nhận ra quy luật chung. Điều này cực kỳ hiệu quả để phân biệt cỏ và cây có màu sắc giống hệt nhau.
- **Đầu ra (Output):** Một bản đồ nhị phân (Binary Mask) kích thước bằng ảnh gốc, trong đó: Pixel Cỏ = 1, Pixel khác = 0.

Module 2: Tăng tốc phần cứng (Hardware Acceleration - The Key)

Một model PyTorch thông thường chạy trên Jetson Nano chỉ đạt khoảng 5-8 FPS (quá chậm).

- **Giải pháp:** Sử dụng **NVIDIA TensorRT**.
 - Chuyển đổi trọng số mô hình từ FP32 (32-bit floating point) sang **FP16** (16-bit). Việc này giảm một nửa lượng RAM tiêu thụ và tăng tốc độ suy luận lên gấp 3-4 lần mà hầu như không làm giảm độ chính xác IoU.
 - *Kết quả kỳ vọng:* Đạt > 30 FPS trên Jetson Nano/Xavier.

Module 3: Ánh xạ không gian (Spatial Mapping & Homography)

Camera nhìn thấy cỏ ở tọa độ pixel (u, v) , nhưng robot cần biết tọa độ mét (x, y) trên mặt đất để di chuyển vòi phun.

- **Ma trận Homography (H):**

- Thực hiện bước **Camera Calibration** (hiệu chỉnh camera) dùng bàn cờ vua để tìm ma trận H .
- Công thức chuyển đổi:

$$xy1 = H \cdot uv1$$

- **Bù trễ (Latency Compensation):**

- Vì mất khoảng 50ms để xử lý ảnh, khi ra lệnh phun thì xe đã chạy thêm được một đoạn.
- Hệ thống cần tính toán vận tốc xe V để phun đúng đắn: $x_{phun} = x_{detect} + V \times \Delta t_{delay}$.

6. Kế hoạch thực nghiệm (Implementation Plan)

6.1. Thiết lập phần cứng (Hardware Setup)

Sinh viên không nhất thiết phải chế tạo cả xe robot (quá sức). Có thể làm mô hình giả lập:

- **Computing Unit:** NVIDIA Jetson Nano (hoặc Xavier NX nếu có điều kiện).
- **Camera:** Camera góc rộng (Wide-angle) hoặc Intel RealSense (để lấy độ sâu).
- **Actuator (Giả lập):** Một dàn đèn LED tương ứng với các vòi phun. Nếu phát hiện cỏ ở vùng nào, đèn LED vùng đó sáng lên.

6.2. Dữ liệu & Huấn luyện

1. **Giai đoạn 1 (Pre-training):** Huấn luyện SegFormer trên dataset **Sugar Beets 2016** hoặc **CWFID**.
2. **Giai đoạn 2 (Fine-tuning):**
 - Tự thu thập dữ liệu nhỏ tại vườn thực nghiệm của trường (khoảng 200 ảnh).
 - Sử dụng công cụ gán nhãn bán tự động (như **CVAT** hoặc **Roboflow**) để tô màu vùng cỏ (Annotation).

6.3. Metrics đánh giá (Độ đo)

- **mIoU (mean Intersection over Union):** Chỉ số quan trọng nhất của bài toán phân đoạn. Mục tiêu $> 80\%$ cho lớp Cỏ dại.
- **Inference Speed (FPS):** Phải đo trực tiếp trên mạch nhúng (Jetson), không đo trên PC.
- **Spraying Accuracy:** Tỉ lệ phun trúng đích (Giả lập).

$$Acc = \frac{Số lần đe nsa'ng đú'ng vị trí cỏ}{Tổng số cụm cỏ}$$

7. Tài nguyên kỹ thuật & Mã nguồn (Resources)

A. GitHub Repositories (Code nền tảng):

1. **MMSegmentation + TensorRT:**

- github.com/open-mmlab/mmdetect
- Đây là công cụ chính chủ để chuyển đổi model từ MMSegmentation sang TensorRT. Rất mạnh mẽ.

2. **Ros_numpy:** github.com/eric-wieser/ros_numpy

- Thư viện giúp chuyển đổi ảnh từ ROS message sang Numpy array để xử lý bằng OpenCV/PyTorch.

B. Công cụ hỗ trợ:

- **CVAT (Computer Vision Annotation Tool):** Công cụ gán nhãn segmentation tốt nhất hiện nay (online/offline).
- **Gazebo Simulator:** Phần mềm giả lập môi trường Robot. Nếu không có robot thật, sinh viên có thể tạo một ruộng ngô ảo trong Gazebo và cho robot ảo chạy để test thuật toán.

Lời khuyên:

1. **Chiến thuật “Cắt gốc” (Crop & Resize):**

- Camera robot thường có độ phân giải cao, nhưng model chỉ cần đầu vào 512×512 .
- Đừng resize toàn bộ ảnh (làm ảnh bị bé đi). Hãy **Crop** (cắt) vùng ngay trước mũi xe (Region of Interest) để xử lý. Vừa nhanh, vừa giữ được chi tiết lá cỏ.

2. **Xử lý ánh sáng (Shadows):**

- Bóng râm của chính con robot in xuống mặt đất thường làm model bị nhiễu.
- *Giải pháp:* Lắp thêm đèn LED trợ sáng ngay cạnh camera và dùng tấm che (shroud) để tạo môi trường ánh sáng ổn định, bất kể trời nắng hay mưa.