**TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

ĐỒ ÁN:

**THỊ GIÁC MÁY TÍNH**

*Đề Tài: Phân đoạn ảnh giao thông*

A blue and white logo

Description automatically generatedGVHD:  Nguyễn Đình Quý

Lớp: 67CS2

Nhóm: 7

Thành viên : Dương Thái Sơn-0328167

Vũ Ngọc Hân -0051467

Phạm Minh Tuyên-0287767

Nguyễn Minh Trí-0246667

1. **Giới thiệu đề tài**

Trong những năm gần đây, với sự phát triển mạnh mẽ của trí tuệ nhân tạo, các kỹ thuật học sâu (Deep Learning) đã được ứng dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là trong xử lý ảnh và thị giác máy tính (Computer Vision). Một trong những bài toán quan trọng và mang tính ứng dụng cao trong lĩnh vực này là **phân đoạn ảnh (Image Segmentation)** – quá trình phân chia ảnh thành các vùng có ý nghĩa, giúp máy tính hiểu và nhận biết các đối tượng trong ảnh một cách chính xác hơn.

Trong bối cảnh giao thông đô thị ngày càng phức tạp, nhu cầu tự động hóa và thông minh hóa các hệ thống hỗ trợ lái xe, giám sát giao thông, và các phương tiện tự hành đang ngày càng tăng cao. Do đó, **phân đoạn ảnh giao thông** – cụ thể là nhận diện và phân biệt chính xác các thực thể như làn đường, xe cộ, người đi bộ, biển báo, đèn tín hiệu,... từ hình ảnh thu được bởi camera – đóng vai trò then chốt trong việc xây dựng các hệ thống giao thông thông minh.

Đề tài **"Phân đoạn ảnh giao thông bằng Deep Learning"** tập trung nghiên cứu và áp dụng các mô hình học sâu hiện đại (như UNet, DeepLab, hoặc UNet++) để thực hiện phân đoạn ảnh giao thông. Dữ liệu đầu vào là các hình ảnh giao thông thực tế (ví dụ từ bộ dữ liệu CamVid hoặc Cityscapes), đầu ra là mặt nạ (mask) phân vùng ảnh tương ứng với từng lớp đối tượng.

Mục tiêu của đề tài không chỉ là huấn luyện mô hình đạt độ chính xác cao mà còn tối ưu hóa hiệu năng mô hình để có thể ứng dụng trong thời gian thực. Bên cạnh đó, đề tài cũng hướng đến việc đánh giá, so sánh hiệu quả giữa các kiến trúc mạng khác nhau trong bài toán phân đoạn ngữ nghĩa ảnh giao thông.

1. **Đặt vấn đề**

Trong bối cảnh đô thị hóa nhanh chóng và số lượng phương tiện giao thông ngày càng gia tăng, Việt Nam đang phải đối mặt với nhiều thách thức về quản lý và đảm bảo an toàn giao thông. Các vấn đề như ùn tắc giao thông, tai nạn giao thông, và ý thức tham gia giao thông chưa cao đang đặt ra yêu cầu cấp thiết về việc hiện đại hóa hạ tầng giao thông, đặc biệt là ứng dụng các công nghệ thông minh vào giám sát và điều hành giao thông.

Hiện nay, một số thành phố lớn như Hà Nội, TP. Hồ Chí Minh, Đà Nẵng đã bắt đầu triển khai các hệ thống camera giám sát, trung tâm điều hành giao thông thông minh,... Tuy nhiên, phần lớn các hệ thống này chưa được tích hợp các công nghệ xử lý ảnh nâng cao để nhận diện và phân tích tự động các tình huống giao thông phức tạp, mà chủ yếu vẫn dựa vào con người hoặc các thuật toán đơn giản.

Trong khi đó, các mô hình học sâu (Deep Learning), đặc biệt là các kiến trúc phân đoạn ảnh như UNet, DeepLab, UNet++,... đã chứng minh hiệu quả vượt trội trong việc nhận diện và phân vùng các đối tượng trong ảnh giao thông. Việc áp dụng các mô hình này không chỉ giúp phát hiện phương tiện, làn đường, người đi bộ hay biển báo mà còn hỗ trợ xây dựng nền tảng cho các hệ thống cảnh báo sớm, xe tự hành hoặc điều phối giao thông tự động.

Do đó, đề tài **“Phân đoạn ảnh giao thông bằng Deep Learning”** có ý nghĩa thực tiễncao tại Việt Nam, góp phần vào xu hướng chuyển đổi số trong lĩnh vực giao thông vận tải. Việc nghiên cứu và áp dụng thành công các mô hình phân đoạn ảnh sẽ là bước đệm quan trọng để phát triển các giải pháp giao thông thông minh, phù hợp với điều kiện thực tế và đặc thù hạ tầng tại nước ta.

1. **Dataset**

**Dataset**

CamVid (Cambridge-driving Labeled Video Database) là một bộ dữ liệu chuẩn được sử dụng rộng rãi trong các nghiên cứu liên quan đến thị giác máy tính, đặc biệt là trong bài toán phân đoạn ngữ nghĩa ảnh giao thông đô thị. Bộ dữ liệu được công bố bởi đại học Cambridge và được trích xuất từ các đoạn video ghi hình trong môi trường giao thông thực tế ở Vương quốc Anh.

1. Đặc điểm chính

* Số lượng ảnh

Khoảng **701 ảnh được gán nhãn thủ công**, chia thành:

* 367 ảnh huấn luyện
* 101 ảnh kiểm tra
* 233 ảnh test
* Số lớp: 12 lớp
* Sky: Bầu trời
* Building: Tòa nhà
* Pole: Cột
* Road: Đường
* Pavement: Vỉa hè
* Tree: Cây
* Sign Symbol: Biển báo
* Fence: Hàng rào
* Car: Ô tô
* Pedestrian: Người đi bộ
* Bicyclist: Người đi xe đạp
* Unlabled: Không xác định
* Định dạng dữ liệu: Dữu liệu bao gồm
* Ảnh RGB gốc
* Mặt nạ phân vùng (segmentation mask) có màu theo từng lớp

1. Ưu điểm

* Là một trong những dataset đầu tiên được gán nhãn đầy đủ cho ảnh video giao thông thực tế.
* Có độ phân giải cao, phù hợp với các mô hình học sâu hiện đại.
* Có cấu trúc đơn giản, dễ sử dụng, và tương thích với nhiều mô hình phân đoạn như UNet, DeepLab, SegNet,...

1. Hạn chế

* Số lượng ảnh tương đối nhỏ so với các bộ dữ liệu hiện đại như Cityscapes, KITTI.
* Cảnh giao thông có thể không đại diện hoàn toàn cho điều kiện ở các quốc gia khác như Việt Nam.
* Không có nhiều biến thiên về thời tiết hoặc điều kiện ban đêm.

1. **Xử lý dữ liệu**
2. **Tiền xử lý ảnh**
3. Bảng ánh xạ màu ↔ lớp (color map)

**color\_map**: ánh xạ từ màu RGB → **class\_id** (0 đến 11)

**id\_to\_color:** ánh xạ ngược từ **class\_id** → màu RGB

1. Chuyển đổi mask

**color\_to\_integer\_mask**: Chuyển mask màu (RGB) → mask số nguyên (class\_id).

**integer\_mask\_to\_color**: Chuyển mask số nguyên → mask màu để hiển thị.

Mục tiêu: dễ dàng train mô hình (bằng mask số nguyên), đồng thời lưu mask màu để trực quan hoá kết quả.

1. Pipeline tiền xử lý bằng Albumentations

A computer screen with text and numbers

AI-generated content may be incorrect.

Thực hiện các phép biến đổi ảnh và mask song song:

* **Resize:** Chuẩn hóa kích thước ảnh về 256 x 256
* **Flip:** Ảnh có tỉ lệ bị lật ngang 50% để tăng tính đa dạng của dữ liệu
* **BrightnessContrast:** Điều chỉnh độ sáng và độ tương phản
* **Normalize:** Chuẩn hóa pixel ảnh RGB từ [0, 255] → về dạng phân phối chuẩn có

**Mean (trung bình)** ≈ 0

**Std (độ lệch chuẩn)** ≈ 1

1. Hàm preprocess\_image

* Chuyển ảnh BGR → RGB.
* Chuyển mask màu thành số nguyên.
* Áp dụng pipeline Albumentations.
* Trả về ảnh đã normalize và mask số nguyên.

1. Hàm process\_dataset

* Duyệt toàn bộ ảnh trong thư mục (train, val, test).
* Ghép với tên file mask tương ứng (\_L).
* Đọc ảnh và mask → xử lý bằng preprocess\_image.
* Giải chuẩn hoá (de-normalize) ảnh để lưu lại ảnh đẹp.
* Chuyển mask số nguyên → mask màu để hiển thị rõ.
* Lưu kết quả ra thư mục đích.

1. **Chia dữ liệu và tạo Dataloader**
2. Mục đích

* Đọc ảnh và mask đã tiền xử lý từ thư mục Camvid\_processed.
* Chuyển **mask màu RGB → mask nhãn (label)** dạng số nguyên (0 đến 11).
* Kết hợp ảnh và mask để lưu dưới dạng .npy nhanh chóng, thuận tiện khi huấn luyện.
* Tạo **PyTorch Dataset và DataLoader** phục vụ huấn luyện mô hình segmentation (UNet).

1. Chia dữ liệu

* **rgb\_to\_label(mask\_rgb, color\_map)**

Chuyển mask RGB (ảnh màu) thành mask nhãn số nguyên **(class\_id).**

Mỗi màu **(R,G,B)** được ánh xạ sang một lớp trong **color\_map.**

* **load\_images\_as\_array(image\_dir)**

Đọc ảnh màu từ thư mục, chuyển thành RGB, và lưu vào list.

Cuối cùng ghép thành **numpy array** có shape: **(N, H, W, 3)**

* **load\_masks\_as\_label\_array(mask\_dir, color\_map)**

Đọc mask màu từ thư mục → chuyển thành **label mask**

(dạng **(N, H, W),** giá trị từ 0-11).

Dùng lại hàm rgb\_to\_label.

* **load\_processed\_Camvid\_label(...)**

Gọi các hàm trên để load:

Tập train: X\_train, Y\_train

Tập val: X\_val, Y\_val

Tập test: X\_test, Y\_test

Các mask từ màu → chuyển sang nhãn luôn trong bước này

* **np.save(...)**

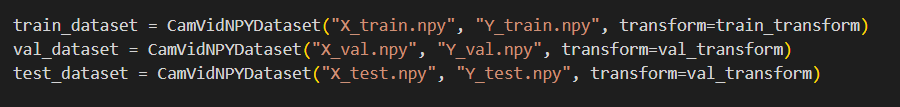
Lưu toàn bộ ảnh và mask thành .npy để:

Load nhanh hơn nhiều so với ảnh .png hoặc .jpg

Giảm chi phí I/O khi training

Dễ kiểm soát batch và shape

1. Tạo Dataset và Dataloader

****

Dùng class CamVidNPYDataset để đọc ảnh/mask từ .npy file.

**transform** sẽ apply **Albumentations augmentation** như định nghĩa

A black background with white text

AI-generated content may be incorrect.

Dùng PyTorch DataLoader để chia mini-batch, giúp huấn luyện hiệu quả.

* Mục đích chính khi sử dụng **Dataloader**

|  |  |
| --- | --- |
| **Mục đích** | **Ý nghĩa** |
| Chia batch | Học theo mini-batch tránh quá tải bộ nhớ |
| Shuffle dữ liệu | Giúp mô hình học tổng quát hơn |
| Load ảnh song song | Tăng tốc độ huấn luyện |
| Kết hợp với transform | Biến đổi ảnh/mask dễ dàng |
| Hỗ trợ train và evaluate | Tiện cho cả train và validation/test |

1. **Kiến trúc mô hình U-net**
2. **Cấu trúc tổng thể**

Unet gồm 2 phần chính:

**Encoder:** Trích xuất đặc trưng và giảm dần kích thước ảnh

**Decorder:** Tăng kích thước ảnh và kết hợp với đặc trưng từ encoder (skip conection) để phcuj hồi độ phân giải

1. **Chi tiết các thành phần**

* **conv\_block(in\_c, out\_c)**

Là khối xử lý cơ bản, gồm:

2 lớp Conv2D (kernel size = 3, padding = 1 → giữ nguyên kích thước).

Kèm BatchNorm và ReLU.

Tăng độ phi tuyến và giúp mô hình học đặc trưng phức tạp

* **Encoder**

**A screen shot of a computer code

AI-generated content may be incorrect.**

Mỗi block encoder đi kèm với MaxPool2d(2) để giảm kích thước ảnh (downsample).

Tăng dần số lượng kênh (feature map)

* **Bottleneck**

**A black background with white text

AI-generated content may be incorrect.**

Là trung tâm "cổ chai", nơi đặc trưng tổng hợp sâu nhất.

* **Decoder**

**A screen shot of a computer program

AI-generated content may be incorrect.**

Mỗi tầng upsampling gồm:

ConvTranspose2d: tăng kích thước ảnh (upsample).

torch.cat(...): ghép đặc trưng từ encoder tương ứng (skip connection).

conv\_block: xử lý đặc trưng đã ghép.

* **Final**

****

Dùng để giảm số kênh về đúng số lớp phân đoạn (12).

1. **Hàm forward()**

* Thực hiện theo thứu tự

Encode → Bottleneck → Decode.

Ghép đặc trưng từ encoder (skip connection) tại mỗi tầng decode để giữ thông tin không gian (biên, cạnh).

* Trả về theo thứ tự **self.final(d1)** là output có shape **[B, C, H, W]**

1. **Hàm mất mát sử dụng trong huấn luyện**
2. **Dice Loss**

* Mục đích

Đo độ trùng giữa vùng dự đoán và vùng thực tế trong segmentation.

Được thiết kế để giải quyết vấn đề mất cân bằng lớp.

* Lý thuyết nền

Dựa trên chỉ số Dice (Dice Coefficient), tương tự như F1-Score

Dice Score =

* Ý nghĩa

Đo lường mức độ chồng lấp giữa vùng segmentation mô hình dự đoán và nhãn thật.

Thường được dùng trong y học, ảnh vệ tinh, nơi vùng quan tâm rất nhỏ.

* Ưu điểm

Miễn nhiễm với mất cân bằng lớp.

Nhấn mạnh vùng nhỏ, vùng quan trọng.

Hiệu quả trong các bài toán vùng hẹp như: khối u, vết nứt, ký hiệu giao thông nhỏ.

* Nhược điểm

Phức tạp hơn CrossEntropy vì cần xử lý softmax và one-hot thủ công.

Không ổn định ở đầu quá trình training, dễ bị dao động.

Dễ gây lỗi khi không có đối tượng (vùng) nào trong nhãn.

1. **CrossEntropy**
   * + Mục đích:

CrossEntropy Loss được sử dụng để đánh giá độ lệch giữa phân phối xác suất dự đoán của mô hình và nhãn thực tế (ground truth). Đây là hàm mất mát chuẩn cho các bài toán phân loại, bao gồm phân đoạn ảnh (mỗi pixel là một mẫu phân loại).

* + - Lý thuyết nền

Dựa trên lý thuyết entropy trong xác suất thông tin: entropy càng nhỏ, dự đoán càng gần đúng.

Kết hợp với softmax để tính xác suất cho từng lớp → so sánh xác suất lớp đúng với nhãn thực tế.

Loss =

Với C là số lớp là mô hình dự đoán cho lớp i

* + - Ý nghĩa

Giúp mô hình tối ưu hóa xác suất dự đoán chính xác lớp của mỗi pixel.

Là cơ sở để đào tạo hầu hết các mô hình phân loại và segmentation

* + - Ưu điểm

Dễ cài đặt, tích hợp sẵn trong PyTorch.

Rất ổn định trong huấn luyện.

Tốt khi các lớp có số lượng tương đương (cân bằng).

* + - Nhược điểm

Không quan tâm đến hình dạng vùng phân đoạn (chỉ xét pixel).

Không hiệu quả khi gặp mất cân bằng lớp nghiêm trọng (class imbalance).

1. **Kết hợp Dice Loss và CrossEntropy**
   * + Mục đích

Tận dụng ưu điểm cả hai hàm mất mát:

CE: chính xác từng pixel.

Dice: chính xác toàn vùng.

Tăng hiệu quả segmentation với vùng nhỏ hoặc mất cân bằng lớp.

* + - Lý thuyết nền

Loss tổng hợp

Combined\_Loss = α⋅CE Loss+β⋅Dice Loss

Mô hình chọn: α=0.7, β=0.3.

* + - Ý nghĩa

Kết hợp giúp mô hình cân bằng giữa học chi tiết pixel và tổng thể vùng.

Cải thiện độ chính xác trong cả vùng lớn lẫn vùng nhỏ.

* + - Ưu điểm

Cải thiện khả năng nhận diện vùng mỏng, nhỏ hoặc hiếm gặp.

Được chứng minh hiệu quả trong segmentation y học, giao thông, tự lái.

* + - Nhược điểm

Cần chọn trọng số hợp lý

Phức tạp hơn trong triển khai, cần chú ý chuẩn hóa, softmax, one-hot encoding.

1. **Các phương pháp đánh giá**
2. **Pixel Accuracy (Độ chính xác điểm ảnh)**

* Công thức:

Pixel Accuracy =

* Cách hoạt động:
* So sánh từng pixel giữa ảnh dự đoán (preds) và ảnh thật (masks).
* Nếu giống nhau thì tính là đúng, khác là sai
* Đếm số pixel đúng, chia cho tổng số pixel.

1. **Mean IoU (Mean Intersection over Union)**

* Công thức

IoU =

Mean IoU = Trung bình IoU trên tất cả các lớp

* Cách hoạt động
* Với từng class, tính vùng giao và vùng hợp giữa vùng được dự đoán và vùng thực tế.
* Sau đó tính trung bình trên tất cả các lớp.

1. **Dice Score (Dice Coefficient / F1 Score)**

* Công thức

Dice Score =

* Cách hoạt động
* Nhấn mạnh vùng giao nhau bằng cách nhân đôi giao.
* Rất gần với F1-score trong phân loại nhị phân.

1. **Thực nghiệm**
2. **Cấu hình chung của các thực nghiệm**

* **Mô hình**: UNet chuẩn (encoder-decoder), số lớp đầu ra: 12
* **Loss Function**: Kết hợp CrossEntropyLoss + DiceLoss
* **Optimizer**: Adam, LR = 1e-4
* **Batch size**: 16
* **Epochs**: 100
* **EarlyStopping**: theo val\_loss
* **Metric theo dõi**: Loss, Pixel Accuracy, Mean IoU, Dice Score
* **Augmentation**: Resize, HorizontalFlip, BrightnessContrast, Normalize (chuẩn ImageNet)

1. **Tổng quan kết quả các lần thực nghiệm**

* **Lần 1**

**A group of graphs showing different colored lines

AI-generated content may be incorrect.**

Xuất hiện hiện tượng bất thường tại epoch ~70:

Loss tăng đột biến, sau đó giảm lại.

Accuracy, mIoU, Dice cũng bị tụt nhẹ rồi hồi phục.

**Nhận xét:** Có dấu hiệu overfitting nhẹ, không ổn định ở cuối quá trình

* **Lần 2**

**A group of graphs showing different colored lines

AI-generated content may be incorrect.**

Biểu đồ loss mượt, chênh lệch train/val rất nhỏ → generalize tốt.

Accuracy tăng đều và ổn định.

Mean IoU và Dice Score tăng liên tục, đạt ~0.52 và ~0.59.

**Nhận xét:** Học ổn định, hiệu suất ổn. Phù hợp để chọn mô hình final.

* **Lần 3**

**A graph of a graph

AI-generated content may be incorrect.**

Train & Val Loss giảm đều, không có overfitting.

Accuracy ổn định quanh ~0.9.

Mean IoU đạt ~0.53 → cho thấy mô hình học tốt ranh giới.

Dice Score đạt ~0.6 → mô hình học rõ vùng đối tượng nhỏ.

**Nhận xét**: Kết quả rất tốt, học ổn định và tiến bộ qua từng epoch. Phù hợp để chọn làm mô hình final

1. **Nhận xét mô hình**
2. **Tính chính xác của mô hình**

Mô hình UNet được huấn luyện trên tập dữ liệu phân đoạn CamVid đã đạt kết quả tốt với các chỉ số như:

* **Pixel Accuracy** cao → cho thấy mô hình phân loại đúng phần lớn các pixel.
* **Mean IoU** và **Dice Score** ở mức ổn định → thể hiện độ chồng khớp tốt giữa vùng dự đoán và vùng thực tế.

1. **Hiệu quả khi kết hợp CrossEntropy và DiceLoss**

* Việc kết hợp cả hai hàm mất mát giúp mô hình:

Phân loại chính xác từng pixel (nhờ CrossEntropyLoss).

Giữ được hình dạng và ranh giới vùng đối tượng (nhờ DiceLoss).

* Tránh được hiện tượng mô hình dự đoán toàn background hoặc bỏ qua lớp nhỏ như người đi bộ, biển báo.

1. **Tính ổn định và khả năng hội tụ**

* Mô hình hội tụ nhanh, loss giảm đều qua các epoch.
* Nhờ sử dụng EarlyStopping theo mIoU, mô hình tránh được tình trạng overfitting, và giữ lại checkpoint tốt nhất.

1. **Khả năng tổng quát trên tập test**

* Khi kiểm tra trên tập test, mô hình vẫn cho kết quả tốt, dự đoán đúng vùng đối tượng quan trọng.
* Tuy nhiên, vẫn còn một số lỗi ở viền rìa đối tượng, đặc biệt là ở những lớp có diện tích nhỏ hoặc xuất hiện hiếm.

1. **Điểm mạnh**

* Mô hình đơn giản, dễ huấn luyện, phù hợp cả trên CPU hoặc GPU.
* Dễ mở rộng: có thể thay backbone hoặc tích hợp với mô hình UNet++.

1. **Hạn chế**

* Vẫn còn sai sót với các đối tượng nhỏ hoặc có ranh giới không rõ ràng.
* Chưa xử lý tốt các trường hợp mờ, nhiễu hoặc ánh sáng phức tạp nếu không tăng cường dữ liệu đầy đủ.

1. **Hướng phát triển**
2. **Nâng cấp kiến trúc mô hình**

* **Thay UNet bằng UNet++:** cải thiện khả năng tái sử dụng đặc trưng và chi tiết ranh giới đối tượng.
* **Kết hợp Attention Mechanism**: tích hợp mô-đun attention (như SE-Block, CBAM) để giúp mô hình tập trung vào vùng quan trọng.
* **Thử nghiệm backbone nâng cao** như ResNet34, EfficientNet hoặc MobileNet để tăng độ sâu và khả năng trích xuất đặc trưng.

1. **Cải tiến hàm mất mát**

* **Focal Loss**: giúp mô hình học tốt hơn với những lớp hiếm và giảm ảnh hưởng từ lớp phổ biến.
* **Tversky Loss**: phiên bản tổng quát của Dice Loss, thích hợp khi dữ liệu mất cân bằng cực cao.
* **Loss động (Dynamic Loss)**: tự động điều chỉnh trọng số giữa CrossEntropy và Dice trong quá trình huấn luyện.

1. **Tăng cường dữu liệu**

* Áp dụng các kỹ thuật nâng cao như:

Elastic Transform (biến dạng co giãn),

GridDistortion,

Random Fog, Snow, hoặc các hiệu ứng thời tiết → tăng độ robust.

* Sinh dữ liệu tổng hợp bằng các mô hình GAN.

1. **Tối ưu hóa quá trình huấn luyện**

* Sử dụng learning rate scheduler như CosineAnnealing, ReduceLROnPlateau.
* Huấn luyện mô hình với Mixed Precision để giảm bộ nhớ và tăng tốc độ.

1. **Mở rộng bài toán**

* Phân đoạn ảnh video **real-time** thay vì ảnh tĩnh.
* Áp dụng mô hình cho các bài toán khác như:

Phân đoạn ảnh y tế (MRI, CT)

Phân đoạn đường phố cho xe tự lái

Phân đoạn ảnh vệ tinh

1. **Kết luận**

Trong đề tài này, chúng em đã xây dựng và huấn luyện thành công mô hình **UNet** cho bài toán **phân đoạn ảnh semantic segmentation** với tập dữ liệu CamVid. Mô hình được tối ưu thông qua việc kết hợp hai hàm mất mát là CrossEntropyLoss và DiceLoss nhằm tận dụng ưu điểm của cả hai: phân loại chính xác từng pixel và giữ được tính toàn vẹn của vùng phân đoạn.

Quá trình huấn luyện mô hình đã đạt được các chỉ số đánh giá tốt như **pixel accuracy, mean IoU, và dice score**, cho thấy mô hình không chỉ dự đoán đúng theo từng lớp mà còn nhận diện chính xác ranh giới các đối tượng trong ảnh. Việc sử dụng kỹ thuật **EarlyStopping** giúp mô hình tránh được overfitting và lưu lại được phiên bản tốt nhất.

Tuy nhiên, mô hình vẫn gặp một số hạn chế khi xử lý các đối tượng nhỏ hoặc vùng phức tạp, điều này mở ra nhiều hướng cải tiến trong tương lai như áp dụng UNet++ hoặc tích hợp attention mechanism.

Tóm lại, mô hình UNet kết hợp với CrossEntropyLoss và DiceLoss là một giải pháp đơn giản nhưng hiệu quả cho các bài toán phân đoạn ảnh, đặc biệt trong môi trường có dữ liệu tương đối cân bằng và yêu cầu độ chính xác cao trong việc phân vùng đối tượng. Những kết quả và kinh nghiệm thu được từ bài toán này là nền tảng quan trọng để phát triển các hệ thống phân đoạn ảnh trong các lĩnh vực như giao thông, y tế, nông nghiệp, và các ứng dụng thị giác máy tính nói chung.