TRƯỜNG ĐẠI HỌC SƯ PHẠM HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

-----📖-----

BÁO CÁO THỰC TẬP

Phát triển Ứng dụng : Computer Vision Object Segmentation

Giảng viên theo dõi:	ThS. Vũ Thái Giang
Sinh viên thực hiện:	Đinh Đức Văn
Lớp:	K69D
Khoa:	Công nghệ thông tin
Mã sinh viên:	695105139

Hà Nội, tháng 03 năm 2024

TRƯỜNG ĐẠI HỌC SƯ PHẠM HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

BÁO CÁO KẾT QUẢ ĐI THỰC TẬP
Thời gian từ 15/11/2024 đến 19/04/2024

Họ và tên sinh viên: Đinh Đức Văn

Lớp: D Điện thoại: 0911662265 Email: ducvan23122000@gmail.com

Cơ sở thực tập: VTI EDUCATION

Tên cơ quan: Công Ty TNHH VTI Education

Địa chỉ: C22BT6 phố Hoài Thanh, khu đô thị Mỹ Đình 2, phường Mỹ Đình 2, quận Nam Từ Liêm, thành phố Hà Nội, Việt Nam

---

Người hướng dẫn: Vũ ĐứcThuận

SDT: 0906282907

Email: thuan.vuduc@vti.com.vn

Nơi công tác: C22BT6 phố Hoài Thanh, khu đô thị Mỹ Đình 2, phường Mỹ Đình 2, quận Nam Từ Liêm, thành phố Hà Nội, Việt Nam

Giáo viên theo dõi (là giáo viên trong khoa): Vũ Thái Giang

SDT 0913040612

Email: giangvt@hnue.edu.vn

Nội dung thực tập:

1. Phát triển Ứng dụng **:** Computer Vision Object Segmentation

Nội dung công việc và kết quả:

2. Problem Description :

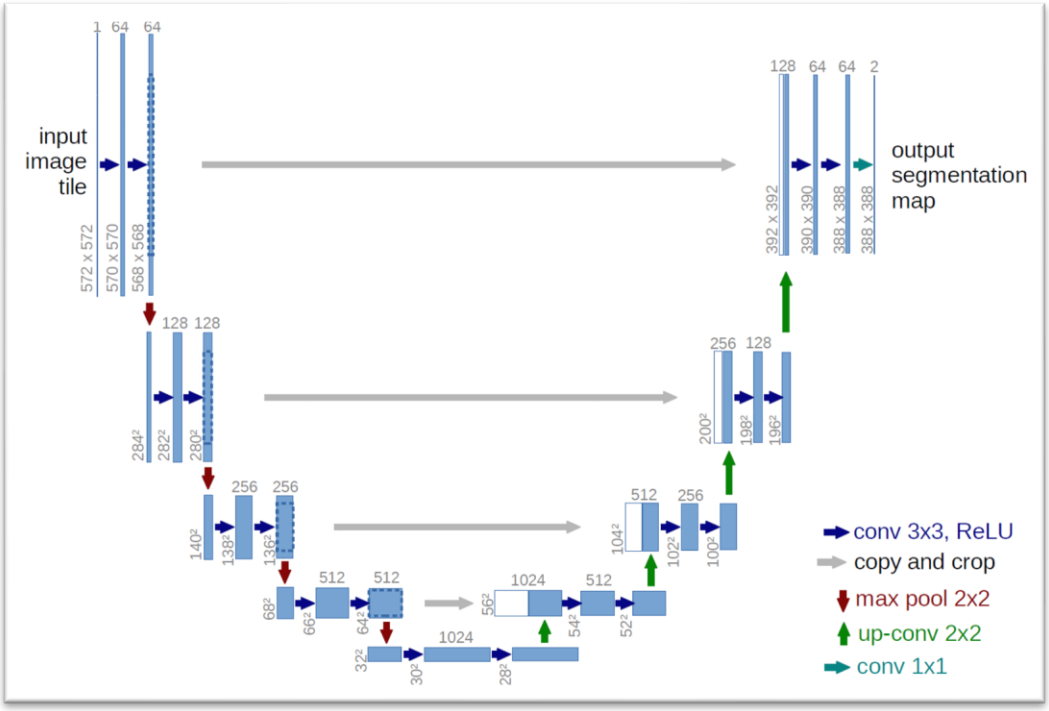
Phân vùng tòa nhà từ ảnh chụp từ trên cao là một nhiệm vụ đầy thách thức vì vật cả từ cây gần đó, bóng của nhà liền kề, kết cấu và màu sắc khác nhau của mái nhà, hình dạng và kích thước khác nhau của các toà nhà là một trong những thách thức khác cản trở các mô hình ngày nay trong việc phân chia ranh giới các tòa nhà sắc nét. Bộ dữ liệu hình ảnh chất lượng cao được chụp từ trên cao tạo điều kiện thuận lợi cho việc so sánh các phương phương pháp hiện có và dẫn đến sự quan tâm ngày càng tăng đối với các ứng dụng hình ảnh trên cao trong cộng đồng máy học và thị giác máy tính.

Dữ liệu Massachusetts Building bao gồm 151 bức ảnh chụp từ trên không của khu vực Boston, với mỗi ảnh có kích thước 1500x1500 của một diện tích ~2.25 Km^1. Do đó, toàn bộ dữ liệu sẽ bao gồm khoảng 340 Km^1. Dữ liệu được chia thành tập train, test và validtation lần lượt là 137, 10, 4 bức ảnh. Bản đồ mục tiêu (nhãn) thu được bằng raster (bitmap) hóa dấu chân tòa nhà thu được từ dự án Open Street Map. Bộ dữ liệu sử dụng hình ảnh do bang Massachusetts công bố, các nhãn dữ liệu được chỉnh sửa thủ công. Tuy nhiên trong dữ liệu có một số bức ảnh bị che và đánh nhãn sai, do đó cần xử lý dữ liệu này.

Hình 1: Dữ liệu huấn luyện ảnh chụp từ trên cao (bên trái), bản đồ mục tiêu (bên phải) hiển thị vùng có buidling (màu trắng).

3. Applied Knowledge :

Xây dựng mô hình học sâu (deep learning): sử dụng kiến trúc UNet (hình 2), các thành phần của mô hình gồm: Conv2d, BatchNrom2d, ReLU, MaxPool2d, ConvTranspose2d. Mô hình sử dụng Conv2d để trích xuất đặc trưng trong ảnh sau đó sử dụng ConvTranspose2d để làm rõ các đặc trưng quan trọng đó lên.



Hình 2: Kiến trúc Unet, mỗi hộp màu xanh tương ứng là một mạng tích chập trích xuất đặc trưng. Số lượng kênh (đặc trưng) đầu ra được hiển thị trên đỉnh của hộp. Hộp màu trắng đại diện cho tính năng sao chép đặc trưng và các mũi tên biểu thị cách hoạt động khác nhau.

- **Conv2d**: Convolution layer là một lớp mạng nơ-ron sâu sử dụng trong xử lý ảnh và thị giác máy tính. Lớp này thực hiện phép tích chập trên ảnh input bằng các bộ lọc qua từng vùng của ảnh, Việc này giúp trích xuất các đặc trưng cục bộ trong ảnh như cạnh, góc, etc. Conv2d giúp giảm số lượng tham số so với một mạng nơ ron tiêu chuẩn và giúp mô hình học được các đặc trưng của ảnh tự động.

- **BatchNorm2d**: là một kỹ thuật chuẩn hóa dữ liệu trong mạng nơ-ron sâu. Kỹ thuật giúp cải thiện tốc độ học và ổn định của mạng bằng cách chuẩn hóa đầu ra của các lớp trước khi áp dụng hàm kích hoạt. BatchNorm giảm hiện tượng “overfitting" và giúp mạng học tốt hơn trong các gia đoạn đầu của quá trình huấn luyện.

- **ReLU**: là một hàm kích hoạt phổ biến được sử dụng trong mạng nơ-ron sâu. Hàm này đơn giản là max(0, x), nghĩa là đầu ra bằng x nếu x là dương và đầu ra bằng 0 nếu x là âm. ReLU giúp tăng tốc độ học và giảm vấn đề biến mất đạo hàm (vanishing gradient) trong quá trình lan truyền ngược.

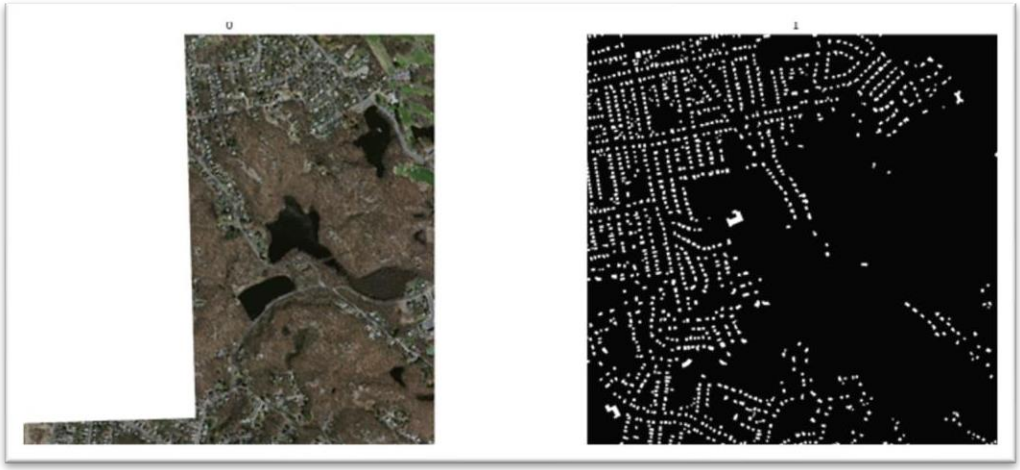
- **MaxPool2d**: là lớp thường được sử dụng sau lớp Conv2d để giảm kích thước đầu ra và trích xuất các đặc trưng quan trọng. MaxPool chia đầu vào thành các vùng không chồng lấn và chọn giá trị lớn nhất từ mỗi vùng. Điều này giúp giảm kích thước đầu ra, giảm tính toán, và giúp mô hình trở nên bất biến với việc dịch chuyển nhỏ trong đầu vào.

- **ConvTranspose2d**: là lớp ngược với Conv2d, thường được gọi là ‘deconvolution'. Lớp này được sử dụng để tăng kích thước đầu ra.

- **Upsample**: Upsample là kỹ thuật tăng kích thước ảnh hoặc dữ liệu bằng cách thêm các giá trị dự đoán giữa các điểm dữ liệu đã có. Upsample thường được sử dụng sau ConvTranspose2d để tăng kích thước của đầu ra và thuận tiện trong việc tạo ra đầu ra với kích thước mong muốn.

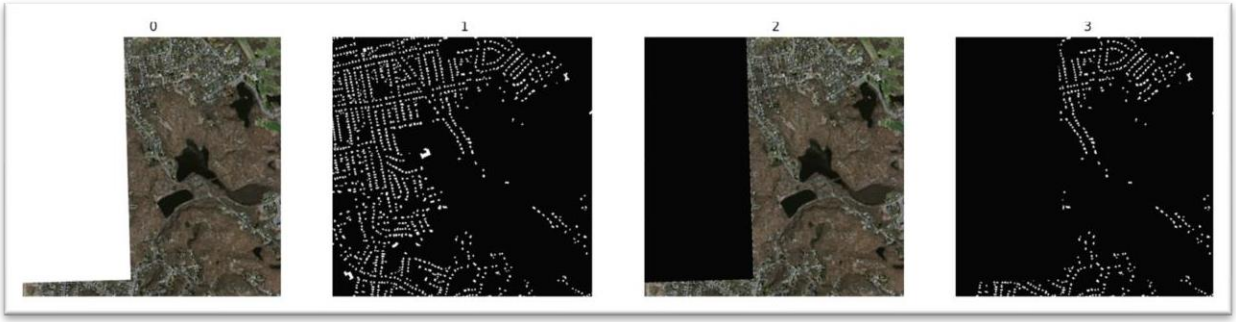
Data Mining: Dữ liệu gồm 151 bức ảnh có kích thước 1500x1500 và nhãn tương ứng. Dữ liệu được chia thành tập train, test, valid lần lượt là 137, 10, 4 (ảnh + nhãn tương ứng). Trong 137 bức ảnh của tập train có một số bức ảnh bị che như hình 3 nhưng nhãn (label) thì không, do đó dữ liệu này đánh nhãn không đúng, nên em đưa ra hai phương án như sau:

- Phương án 1: Loại bỏ các bức ảnh bị che và nhãn tương ứng ra khỏi dữ liệu.



Hình 3:(Dữ liệu sai) Ảnh bị che và nhãn tương ứng thì không bị che

- Phương án 2: Chỉnh sửa nhãn theo vùng bị che của ảnh: để tận dụng tối đa số lượng dữ liệu, thay vì loại bỏ thì sẽ sửa nhãn theo vùng ảnh bị che như hình dưới, cụ thể là vùng bị che sẽ đánh nhãn là “background”.



Hình 4: Hình ảnh bị che một phần và nhãn tương ứng bị sai (hai hình ảnh bên trái), hình ảnh bị che một phần và nhãn tương ứng đã được sửa (hai hình ảnh bên phải).

Github: Trong dự án có sử dụng github để cài đặt một số thư viện để xây dựng các hàm cần thiết như: hàm loss, hàm train, metrix (IoU).
Link github: https://github.com/qubvel/segmentation_models.pytorch.git

Kỹ thuật One-hot (encode và decode): dùng để chuyển bức ảnh nhãn từ màu RGB đen trắng sang ảnh nhị phân (giá trị 0 và 1).

Kỹ thuật làm giàu dữ liệu: dự án sử dụng thư viện "albumentations" để tăng cường dữ liệu như: RandomCrop, HorizontaFlip, VerticalFlip, Rotate90 tương ứng là cắt một vùng ngẫu nhiên trong bức ảnh với kích thước 256x256, và áp dụng ngẫu nhiên trong các phép biến đổi lật ngang, lật dọc hoặc xoay 90 độ.

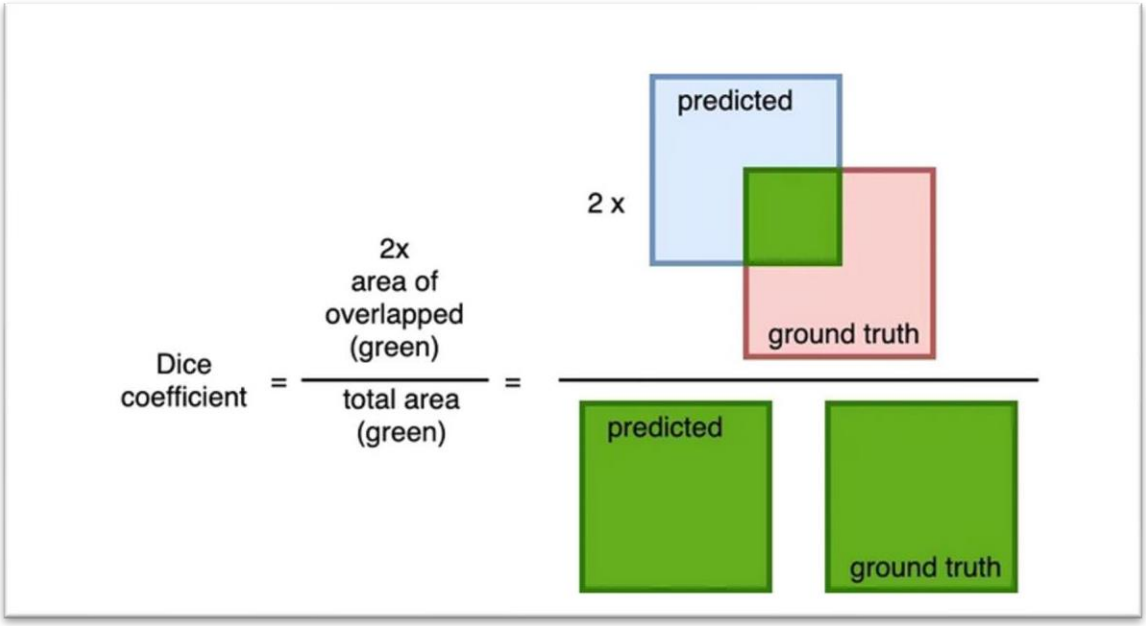
Hàm tính loss: Đối với bài toán segmentation, hàm loss thường được sử dụng là Dice Loss (còn gọi là F1 Loss). Hàm này đo lượng sự tương đồng giữa các vùng được dự đoán và nhãn bằng cách tính tỷ lệ giữa diện tích giao nhau và tổng diện tích của hai vùng (dự đoán và nhãn). Dice Loss thường hoạt động tốt với các trường hợp có sự không cân bằng giữa đối tượng và nền, giúp mô hình tập trung vào việc phát hiện đối tượng một cách chính xác.

Phương trình toán học của Dice Loss:

$$Loss = 1 - \frac{2 \times \sum(P \times G)}{\sum P + \sum G + \varepsilon}$$

Trong đó:

- *P là dự đoán (giá trị 0-1 cho mỗi pixel)*
- *G là nhãn thực tế (0,1)*
- *∑ là phép tổng các pixel*
- *ε là một số rất nhỏ (thường được sử dụng để tránh chia cho mẫu số 0)*



Hình 5: Ảnh mô tả Dice Loss, 2 lần diện tích phần chung chia cho tổng hai diện tích, Dice Loss sẽ tiệm cận 1 khi hai vùng dự đoán và nhãn trùng nhau.

Hàm đánh giá (IoU): IoU sử dụng để đo lường độ chính xác và đánh giá hiệu suất của mô hình segmentation trong quá trình huấn luyện và kiểm tra. Thuật toán đo lường tỷ lệ diện tích phần chung giữa vùng dự đoán và vùng thực tế (đối tượng) so với tổng diện tích của hai vùng. Chỉ số này cung cấp cái nhìn tổng quan về khả năng phân đoạn đúng và chính xác của mô hình.

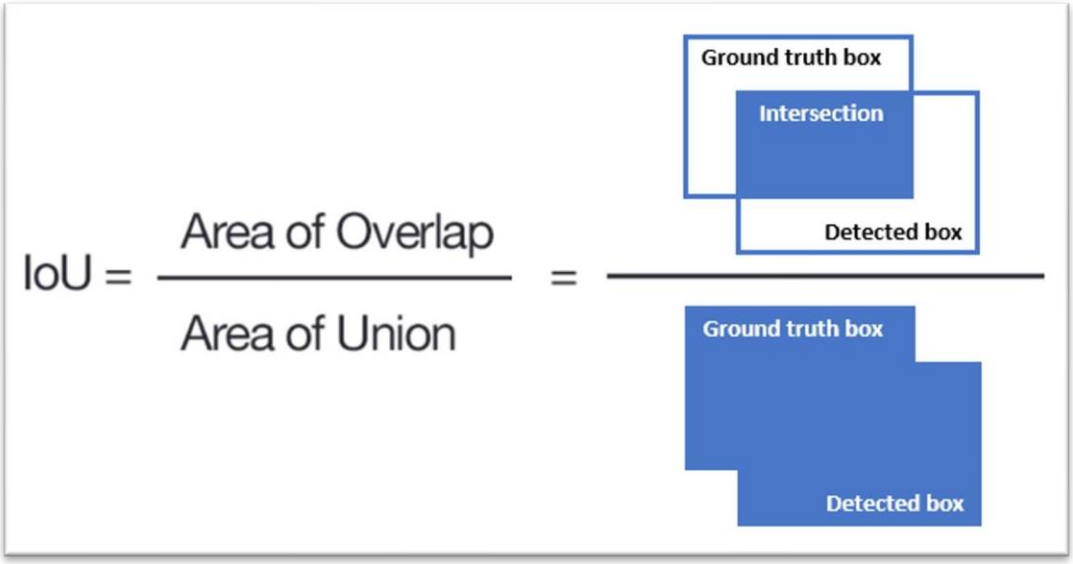
Công thức toán của IoU là:

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} = \frac{TP}{TP + FP + FN}$$

Trong đó:

- *TP (True Positive): Số pixel được mô hình dự đoán là đối tượng và nhãn cũng là đối tượng.*
- *FP (False Positive): Số pixel được mô hình dự đoán là đối tượng, nhưng nhãn thực tế lại không phải đối tượng.*
- *FN (False Negative): Số pixel được mô hình dự đoán không phải đối tượng, nhưng nhãn thực tế là đối tượng.*

Giá trị IoU càng gần 1, tức là có một phần lớn vùng được dự đoán trùng khớp với vùng thực tế (nhãn) và ngược lại dự đoán của mô hình không tương đồng với thực tế nếu IoU gần 0.



Hình 6: Ảnh mô tả hàm IoU là tỷ lệ giữa diện tích chung và tổng diện tích của hai vùng.

Hàm đánh giá mô hình:

- **Accuracy**: tính tỷ lệ pixel đoán đúng trên tổng số pixel của ảnh.
- **Score**: tính tỷ lệ pixel đoán đúng là đối tượng trên tổng số pixcel là đối tượng có trong nhãn.

4. Results :

Trong dự án này, em đã chia ra hai trường hợp như đã nêu ở mục Data Mining để huấn luyện mô hình. Vậy sẽ có hai kết quả để so sánh và nhận xét.

Các mô hình được train 24 epochs với hai trường hợp dữ liệu sau:

- Trường hợp 1: Train mô hình với dữ liệu bỏ các ảnh bị che và nhãn tương ứng
- Trường hợp 2: Train mô hình với dữ liệu được chỉnh sửa nhãn theo vùng bị che của ảnh.

Bảng 1: Kết quả đánh giá mô hình trên tập dữ liệu test gồm 10 bức ảnh và nhãn (3 kết quả hình ảnh ở cuối report).

	Trường hợp 1	**Trường hợp 2**
Mean IoU value	0.75	0.75
Mean Dice Loss value	0.33	0.32
Mean Accuracy value	88.1 %	88.7 %
Mean Score value	71.31 %	72.2 %

Kết luận:

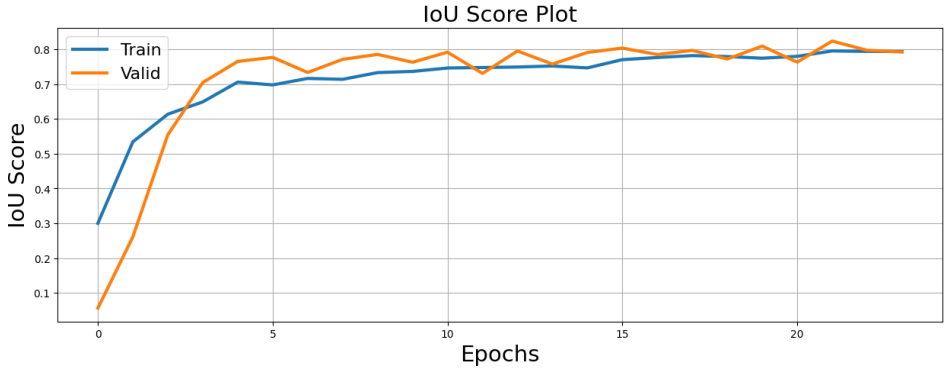
hình UNet đã được đào tạo để có thể dự đoán đúng vị trí, hình dạng và thông tin các điểm ảnh (Mô pixels) thuộc các đối tượng nào trong hình ảnh. Mô hình đã có khả năng nhận biết các ngôi nhà trong ảnh chụp từ trên cao.

Nhận xét hai trường hợp cho thấy trường hợp 2 cho kết quả tố hơn một chút so với trường hợp 1 với hai cách đánh giá là Accuracy và Score. Có thể giải thích vì trường hợp 1 ít hơn 31 bức ảnh để huấn luyện mô hình.

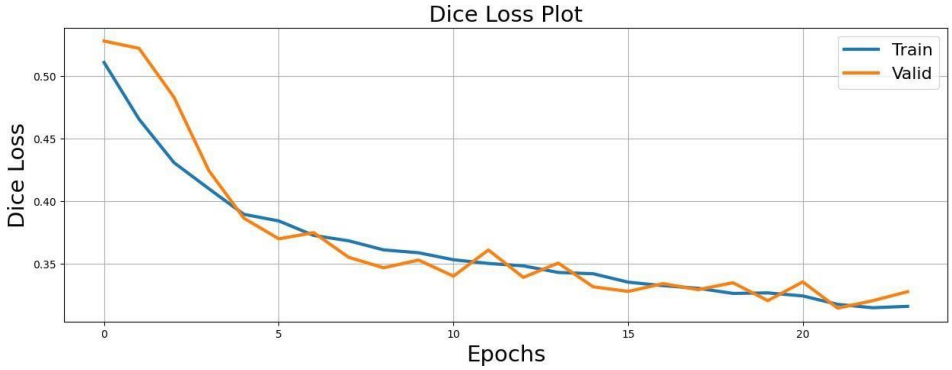
Hiệu suất và độ chính xác của mô hình được thể hiện trong Bảng 1. Để cải thiện kết quả dự đoán, có thể thực hiện một số cải tiến như tăng số lượng lần lặp (epochs) trong quá trình huấn luyện mô hình, cải thiện quá trình khai thác dữ liệu (Data Mining) hoặc thêm dữ liệu mới.

Một số hướng tiếp cận bao gồm việc tối ưu kiến trúc mô hình, nâng cao khả năng khai thác dữ liệu và tăng cường dữ liệu đào tạo.

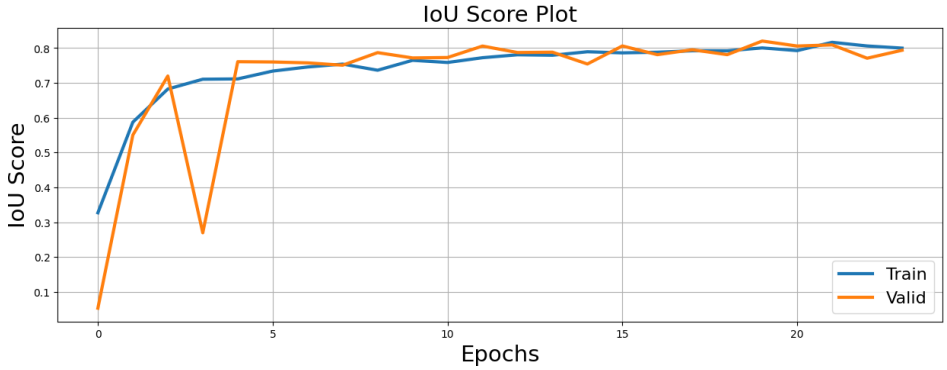
Kết quả đồ thị IoU và Dice Loss với số Epochs trong quá trình train và valid mô hình trên tập dữ liệu train và valid.



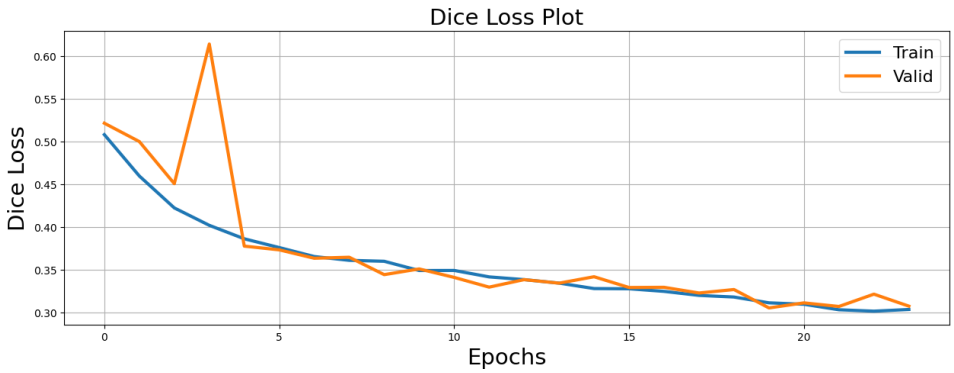
Hình 7: (Trường hợp 1) Giá trị IoU Score với số Epochs đã train và valid của mô hình.



Hình 8: (Trường hợp 1) Giá trị Dice Loss với số Epochs đã train và valid của mô hình

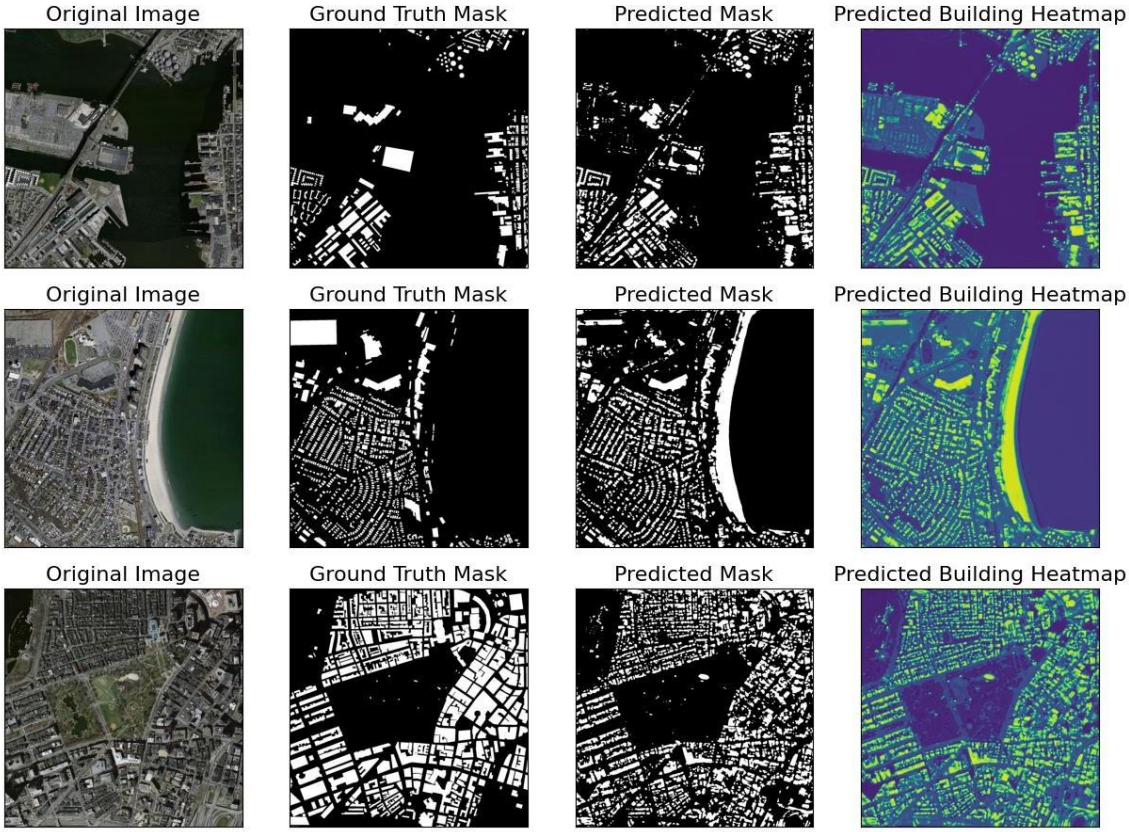


Hình 9: (Trường hợp 2) Giá trị IoU Score với số Epochs đã train và valid của mô hình



Hình 10: (Trường hợp 2) Giá trị Dice Loss với số Epochs đã train và valid của mô hình

Kết quả dự đoán của mô hình (trường hợp 1): 3 ảnh đại diện cho tập test



Kết quả dự đoán của mô hình (trường hợp 2): 3 ảnh đại diện cho tập test