

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



TRẦN XUÂN HÒA

LUẬN VĂN THẠC SĨ KỸ THUẬT

HÀ NỘI - 2020

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



TRẦN XUÂN HÒA

**ĐỀ TÀI: PHÁT HIỆN MẮT RỪNG VEN BIỂN
TỪ ẢNH VỆ TINH DỰA TRÊN MẠNG NƠ-RON**

CHUYÊN NGÀNH HỆ THỐNG THÔNG TIN

MÃ SỐ 08.48.01.04

NGƯỜI HƯỚNG DẪN KHOA HỌC

TS. NGUYỄN TRỌNG KHÁNH

HÀ NỘI - 2020

LỜI CAM ĐOAN

Tôi xin cam đoan luận văn về đề tài ***“Phát hiện mất rừng ven biển từ ảnh vệ tinh dựa trên mạng nơ-ron”*** là công trình nghiên cứu cá nhân của tôi trong thời gian qua. Mọi số liệu sử dụng phân tích trong luận văn và kết quả nghiên cứu là do tôi tự tìm hiểu, phân tích một cách khách quan, trung thực, có nguồn gốc rõ ràng và chưa được công bố dưới bất kỳ hình thức nào.

Tôi xin chịu hoàn toàn trách nhiệm nếu có sự không trung thực trong thông tin sử dụng trong công trình nghiên cứu này.

Hà Nội, ngày 08 tháng 06 năm 2020

Tác giả đề tài

Trần Xuân Hòa

LỜI CẢM ƠN

Để hoàn thành được luận văn, ngoài sự nghiên cứu và cố gắng của bản thân, tôi xin cảm ơn thầy giáo TS. Nguyễn Trọng Khánh - người thầy trực tiếp hướng dẫn, tận tình chỉ bảo và định hướng cho tôi trong suốt quá trình thực hiện luận văn. Một lời cảm ơn chắc chắn không thể diễn tả hết lòng biết ơn sâu sắc của tôi tới thầy một người thầy của tôi trên mọi phương diện!

Tôi xin gửi lời cảm ơn chân thành cảm ơn tất cả các thầy cô giáo của Học viện Công nghệ Bru chính Viễn thông đã giảng dạy, quan tâm nhiệt tình và dìu dắt tôi trong suốt quá trình học tập tại trường.

Cuối cùng, tôi xin gửi lời cảm ơn tới gia đình, bạn bè và những người đã luôn ở bên tôi cổ vũ, động viên, tạo điều kiện thuận lợi cho tôi học tập, tạo động lực tinh thần vô giá để tôi hoàn thiện luận văn này và ngày một hoàn thiện chính bản thân mình.

Trong quá trình nghiên cứu và thực hiện luận văn, mặc dù được sự hướng dẫn nhiệt tình của thầy giáo TS. Nguyễn Trọng Khánh và những nỗ lực của bản thân nhưng cũng không thể tránh khỏi những thiếu sót hạn chế. Tôi rất mong nhận được ý kiến đóng góp, sửa chữa từ quý Thầy, Cô và các bạn bè đồng nghiệp để luận văn được hoàn thiện hơn.

Trân trọng cảm ơn!

Tác giả

Trần Xuân Hòa

MỤC LỤC

MỤC LỤC	i
DANH MỤC HÌNH ẢNH.....	iv
DANH MỤC BẢNG	vi
DANH MỤC TỪ VIẾT TẮT	vii
MỞ ĐẦU	i
1. Tính cấp thiết của đề tài	i
2. Tổng quan về vấn đề nghiên cứu	ii
3. Mục đích nghiên cứu.....	iii
4. Đối tượng và phạm vi nghiên cứu	iii
5. Phương pháp nghiên cứu	iv
6. Nội dung của luận văn	iv
CHƯƠNG 1 TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU	1
1.1 Đặt vấn đề	1
1.2 Tổng quan về rừng ven biển và biến động rừng ven biển.	2
1.2.1 Rừng ven biển.	2
1.2.2 Hiện trạng rừng ven biển Việt Nam.....	3
1.2.3 Nguyên nhân biến động rừng ven biển.	4
1.3 Các nghiên cứu liên quan phát hiện biến động rừng từ ảnh vệ tinh.	5
1.3.1 Khả năng khai thác ảnh vệ tinh phục vụ phát hiện biến động rừng.....	5
1.3.2 Một số kết quả nghiên cứu về phát hiện biến động rừng từ ảnh vệ tinh.....	6
1.3.3 Khó khăn và thách thức trong phát hiện biến động rừng từ ảnh vệ tinh.....	7
1.4 Ứng dụng mạng nơ-ron giải bài toán phát hiện mất rừng từ ảnh vệ tinh.	8
1.5 Một số công nghệ liên quan.....	9
1.5.1 Viễn thám.	9
1.5.2 Ảnh vệ tinh.....	11

1.5.3	Hệ thống thông tin địa lý.....	15
1.5.4	Điện toán đám mây	16
1.6	Tổng kết chương 1	18
CHƯƠNG 2 PHÁT HIỆN MẮT RỪNG VEN BIỂN TỪ ẢNH VỆ TINH DỰA TRÊN MẠNG NƠ-RON		19
2.1	Mạng nơ-ron.	19
2.2	Mạng nơ-ron tích chập.....	19
2.2.1	Giới thiệu tổng quan.....	19
2.2.2	Các kiểu tầng.....	21
2.2.3	Các hàm kích hoạt thường sử dụng.....	23
2.3	Kiến trúc mạng U-Net.....	24
2.3.1	Hiểu về phân loại hình ảnh, phát hiện đối tượng và phân đoạn hình ảnh.....	24
2.3.2	Giới thiệu U-Net.....	24
2.3.3	Kiến trúc mạng U-Net.....	25
2.3.4	Đặc điểm mạng U-Net.	26
2.4	Phát hiện mắt rừng từ ảnh vệ tinh dựa trên mạng U-Net.....	27
2.4.1	Mô tả bài toán.....	27
2.4.2	Quy trình thực hiện.	28
2.5	Tổng kết chương 2	30
CHƯƠNG 3 THỬ NGHIỆM VÀ ĐÁNH GIÁ.....		31
3.1	Phần mềm và thư viện.....	31
3.1.1	Phần mềm.....	31
3.1.2	Thư viện.	32
3.2	Chuẩn bị dữ liệu.....	32
3.2.1	Vị trí thử nghiệm.....	32
3.2.2	Dữ liệu ảnh Sentinel-2.....	33
3.2.3	Dữ liệu bản đồ hiện trạng rừng.	34

3.2.4	Dữ liệu điểm mẫu.	35
3.3	Cài đặt thực nghiệm.	36
3.3.1	Cài đặt thư viện và thiết lập chung.	36
3.3.2	Xây dựng các hàm xử lý.	38
3.3.3	Thực thi mạng U-Net.	42
3.3.4	Huấn luyện mô hình.	45
3.3.5	Dự đoán.	47
3.4	Kết quả thử nghiệm và đánh giá.	49
3.4.1	Kết quả thử nghiệm phương pháp U-Net phát hiện biến động rừng.	49
3.4.2	So sánh kết quả phát hiện biến động giữa phương pháp MCVA và U-Net..	50
3.4.3	Đánh giá kết quả.	53
3.5	Tổng kết chương 3	53
CHƯƠNG 4 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		54
4.1	Những đóng góp của luận văn.	54
4.2	Hướng phát triển.	54
KẾT LUẬN		55
DANH MỤC CÁC TÀI LIỆU THAM KHẢO		56
PHỤ LỤC 1 DANH SÁCH CẢNH ẢNH SENTINEL-2 SỬ DỤNG HUẤN LUYỆN ...		58
PHỤ LỤC 2 HOẠT ĐỘNG ĐẦY ĐỦ CỦA CÁC LỚP TRONG MẠNG U-NET.....		60

DANH MỤC HÌNH ẢNH

Hình 1.1 - Rừng ngập mặn tàn phá sau bão và đê kè biển bị sạt lở do triều cường.	1
Hình 1.2 - Bản đồ các tỉnh, thành phố có rừng ven biển.	2
Hình 1.3 – Bản đồ hiện trạng RVB Việt Nam (31/12/2018).....	3
Hình 1.4 - Ảnh chụp bề mặt trái đất từ vệ tinh Sentinel-2 ngày 04/06/2020	7
Hình 1.5 - Tương tác năng lượng trong khí quyển và trên bề mặt Trái Đất	10
Hình 1.6 - Các dải phổ dùng trong viễn thám.	10
Hình 1.7 - Phản xạ phổ tương ứng với các loại đối tượng	11
Hình 1.8 - Ứng dụng viễn thám trong quản lý đô thị.	11
Hình 1.9 - Vệ tinh Sentinel-2 chụp ảnh bề mặt Trái Đất.....	13
Hình 1.10 - Một số bộ dữ liệu sẵn có trên Google Earth Engine.	17
Hình 1.11 - Mô hình đơn giản hóa hệ thống Google Earth Engine.....	18
Hình 2.1 - Các thành phần cấu thành mạng nơ-ron tích chập (CNN).	20
Hình 2.2 - Cấu trúc đại diện của một mạng nơ-ron tích chập đầy đủ.	21
Hình 2.3 - Ví dụ phép tích chập.	21
Hình 2.4 - Ví dụ max pooling và average pooling.	21
Hình 2.5 – Mô tả tầng kết nối đầy đủ (Fully connected).	22
Hình 2.6 - Hàm softmax	23
Hình 2.7 - Phân loại hình ảnh, phát hiện đối tượng và phân đoạn hình ảnh.	24
Hình 2.8 - Ứng dụng U-Net trong phân đoạn và dự đoán ảnh viễn thám.	24
Hình 2.9 - Kiến trúc mạng U-Net.	25
Hình 2.10 - Mô tả cho một quá trình tích chập và giải mã trong U-Net	26
Hình 2.11 - Phương pháp padding method.....	26
Hình 2.12 - Quy trình phát hiện biến động rừng từ ảnh vệ tinh dựa trên mạng nơ-ron.	29
Hình 3.1 - Giao diện phần mềm Colab.....	31
Hình 3.2 - Bản đồ thành phố Hải Phòng	32

Hình 3.3 - Bản đồ cảnh ảnh sử dụng đã loại bỏ mây và bóng mây.....	33
Hình 3.4 - Bản đồ thể hiện hiện trạng rừng trên nền ranh giới thành phố Hải Phòng.	34
Hình 3.5 - Bản đồ hiển thị các điểm mẫu khóa ảnh huấn luyện và xác minh.	35
Hình 3.6 - Lớp phủ rừng ven biển Hải Phòng	37
Hình 3.7 - Kết quả xử lý ảnh loại bỏ mây	39
Hình 3.8 - Xuất vùng lân cận điểm ảnh.....	40
Hình 3.9 - Dữ liệu huấn luyện được lưu với định dạng .tfrecord trong Google Drive.	40
Hình 3.10 – Mô hình mạng U-Net thử nghiệm.	42
Hình 3.11 - Biểu đồ đường cong quá trình huấn luyện.	46
Hình 3.12 - Kết quả dự đoán ảnh thử nghiệm.	48
Hình 3.13 - Kết quả thử nghiệm, phát hiện biến động mất rừng tại Huyện Thủy Nguyên, TP. Hải Phòng.....	49
Hình 3.14 - Sơ đồ quá trình xử lý, phân tích dữ liệu.....	50
Hình 3.15 - Kịch bản so sánh ở cùng vị trí địa lý giữa phương pháp MCVA và U-Net...	51
Hình 3.16 - So sánh kết quả phát hiện biến động mất rừng giữa phương pháp MCVA và U-Net.	52

DANH MỤC BẢNG

Bảng 1.1 - Số sánh diện tích RVB có rừng năm 2017 và năm 2018.....	4
Bảng 1.2 - Danh sách các kênh ảnh, bước sóng và độ phân giải của ảnh Sentinel-2	12
Bảng 1.3 - Phân loại giá trị NDVI theo chất lượng thực vật trong lớp phủ bề mặt trái đất	14
Bảng 1.4 - Bảng phân loại NDWI theo chất lượng thực vật trong lớp phủ bề mặt trái đất.	15
Bảng 2.1 - So sánh max pooling và average pooling.	22
Bảng 2.2 - So sánh cách biến thể của hàm kích hoạt ReLU.....	23
Bảng 3.1 - Thông kê diện tích hiện trạng RVB TP. Hải Phòng (31/12/2019).	34
Bảng 3.2 - Mẫu thu thập dữ liệu điểm mẫu khóa xác nhận tại thực địa.....	35
Bảng 3.3 - Thông tin huấn luyện mô hình.....	45

DANH MỤC TỪ VIẾT TẮT

BĐKH	Biến đổi khí hậu
RVB	Rừng ven biển
RNM	Rừng ngập mặn
AI	Artificial Intelligence
ML	Machine Learning
NN	Neural Networks
DL	Deep Learning
ANN	Artificial Neural Network
CNN	Convolutional Neural Networks
DNN	Deep Neural Networks
GEE	Google Earth Engine
TF	TensorFlow
RS	Remote Sensing
GIS	Geographic Information System
MLC	Maximum Likelihood classifier
MCVA	Multi-variant Change Vector Analysis
NDVI	Normalized Difference Vegetation Index
NDSI	Normalised Difference Snow Index
NDWI	Normalized Difference Water Index
CONV	Convolution
POOL	Pooling
FC	Fully Connected
ReLU	Rectified Linear Unit

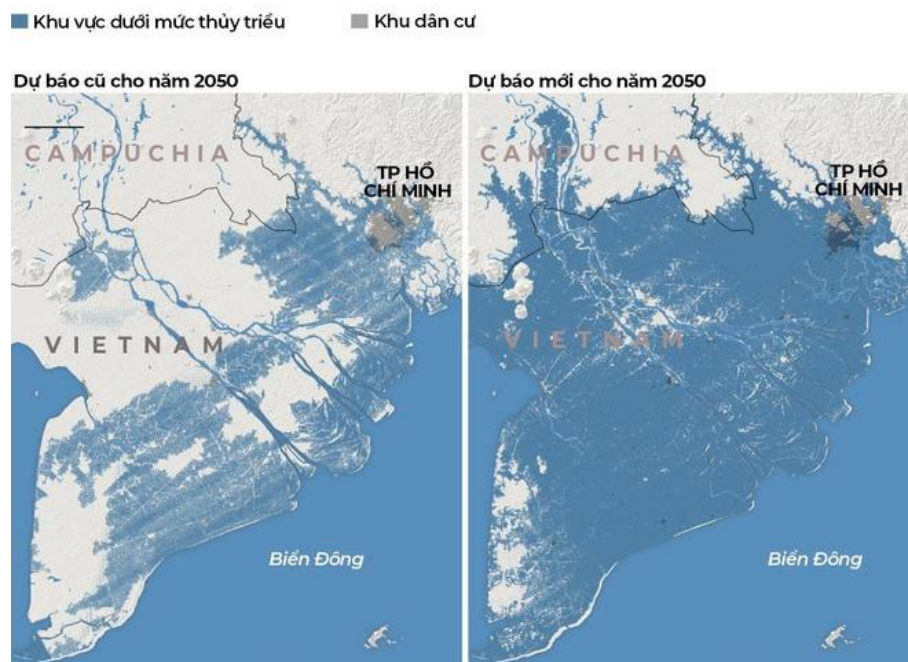
MỞ ĐẦU

1. Tính cấp thiết của đề tài

Biến đổi khí hậu (BĐKH) đang ngày càng phức tạp và báo động. Trái đất liên tục phải chứng kiến nhiều hình thái thời tiết cực đoan bắt nguồn từ BĐKH. Sự nóng lên toàn cầu dẫn đến băng tan, nước biển dâng cao, lũ lụt, hạn hán, cháy rừng...vv kéo theo hệ lụy sinh thái tác động nghiêm trọng đến con người.



Theo những thống kê mới nhất: “Trên toàn cầu, Việt Nam là quốc gia xếp thứ sáu chịu ảnh hưởng nặng nề do BĐKH¹”. Các tác động do BĐKH ảnh hưởng đến hơn 74% dân số. Xu hướng này được dự báo sẽ tăng cường theo thời gian. Đến năm 2050, tình trạng mực nước biển dâng cao có thể làm ảnh hưởng trầm trọng hơn đến các khu vực ven biển và xóa sổ nhiều thành phố lớn ven biển².



Trải dài từ Móng Cái đến Cà Mau, nước ta có trên 3.260 km đường bờ biển và diện tích vùng ven biển rộng lớn. Đây không chỉ là nơi cư trú mà còn là nơi cung cấp nguồn dinh dưỡng, hỗ trợ cho sự tồn tại và phát triển phong phú của các quần thể sinh vật cửa sông ven biển, đồng thời là nơi duy trì đa dạng sinh học cho biển.

¹ Nguồn: [unicef.org/vietnam/vi/trẻ-em-và-biến-đổi-khí-hậu](https://www.unicef.org/vietnam/vi/trẻ-em-và-biến-đổi-khí-hậu)

² Nguồn: [nytimes.com/interactive/2019/10/29/climate/coastal-cities-underwater.html](https://www.nytimes.com/interactive/2019/10/29/climate/coastal-cities-underwater.html)

Vùng ven biển nói chung, Rừng ven biển (RVB) & Rừng ngập mặn (RNM) nói riêng đối với môi trường sinh thái đã được các nhà khoa học khẳng định từ lâu. Được đánh giá là bức tường xanh vững chắc giúp ứng phó với sự BĐKH khi mực nước biển dâng cao, góp phần bảo vệ con người, bảo vệ vùng bờ biển trước tác động từ thiên tai.

Trong những năm qua, diện tích RVB biến động khá nhanh với quy mô ngày càng lớn. Ngoài sự ảnh hưởng của các yếu tố thiên tai thì một phần không nhỏ làm suy giảm diện tích RVB là do yếu tố con người: phát triển vùng kinh tế nóng phá vỡ quy hoạch, vãn nạn di dân tự do chặt phá rừng...vv

Nhưng những hạn chế về nhân lực, trang thiết bị và giải pháp công nghệ hỗ trợ nên việc phát hiện biến động mất rừng tại nhiều nơi là vô cùng khó khăn. Làm hiệu quả phòng chống nạn chặt phá rừng và tính minh bạch trong công tác quản lý bảo vệ thường thấp. Chính vì vậy, việc theo dõi dự báo biến động rừng là rất cần thiết.

Để giải quyết vấn đề trên cần tập trung xây dựng giải pháp xác định những dấu tích thay đổi từ ảnh vệ tinh; ứng dụng trí tuệ nhân tạo trong phân tích xử lý và giải đoán ảnh, giải quyết các bài toán thực tế nêu trên. Điều đó cho thấy khả năng ứng dụng công nghệ cao trong quản lý bảo vệ tài nguyên thiên nhiên, bảo vệ môi trường và chống BĐKH.

Vì những lý do trên, cùng với mong muốn làm chủ công nghệ, mở rộng kiến thức nghiên cứu về trí tuệ nhân tạo ứng dụng cho các bài toán giám sát bảo vệ tài nguyên rừng. Dưới sự hướng dẫn của **TS. Nguyễn Trọng Khánh**, tôi quyết định chọn đề tài nghiên cứu: *“Phát hiện mất rừng ven biển từ ảnh vệ tinh dựa trên mạng nơ-ron”*.

Luận văn sẽ tập trung nghiên cứu giải quyết bài toán phân loại đối tượng, phân vùng đối tượng xác định điểm dị thường trong ảnh dựa trên mạng nơ-ron một cách có hệ thống trên cơ sở lý thuyết vững chắc. Mô hình giải pháp đưa ra sẽ được áp dụng thử nghiệm theo dõi mất rừng tại các khu vực có rừng dọc theo đường bờ biển.

2. Tổng quan về vấn đề nghiên cứu

Cách tiếp cận hiện nay để phát hiện mất rừng có thể được phân chia thành hai cách: (1) xác định thay đổi rừng bằng cách so sánh hình ảnh chụp tại khoảng thời gian tương tự giữa các năm; (2) theo dõi những thay đổi rừng bằng cách sử dụng nhiều hình ảnh chụp trong năm. Có thể thấy khoảng thời gian cần thiết cho những phương pháp tiếp cận truyền thống như vậy là quá dài, làm giảm hiệu quả quản lý bảo vệ rừng và phòng chống nạn chặt phá rừng.

Sử dụng học máy huấn luyện hiểu biết về ảnh vệ tinh có thể cung cấp giải pháp giám sát tốt hơn so với phương pháp truyền thống. Phát hiện và xác định vị trí mất rừng sớm giúp các cơ quan quản lý, chủ sở hữu rừng và các bên liên quan tại địa phương phối hợp phản ứng bảo vệ nhanh và hiệu quả hơn.

Trong nghiên cứu này, tôi thiết kế thử nghiệm một kiến trúc Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) thực hiện huấn luyện, học sâu (Deep Learning - DL) các hình ảnh vệ tinh từ đó có thể đưa ra cảnh báo mất rừng trong tương lai. Đầu vào cho thuật toán gồm: (1) Dữ liệu ảnh vệ tinh được khai thác từ Google Earth Engine; (2) Dữ liệu hiện trạng rừng khai thác từ Hệ thống thông tin quản lý ngành Lâm nghiệp - FORMIS; (3) Dữ liệu điểm mẫu (gồm điểm huấn luyện và điểm mẫu xác minh) kế thừa từ hoạt động điều tra khảo sát thực địa. Mạng CNN học và phân loại dữ liệu đầu vào để dự đoán điều kiện khí quyển, địa hình (đất, núi, sông, rừng, đồng bằng..vv) hoặc các hoạt động của con người. Đầu ra mục tiêu hướng đến là có thể phát hiện nạn phá rừng hoặc rừng bị mất do thiên tai ở những nơi con người không thể tiếp cận dễ dàng ngay lập tức.

3. Mục đích nghiên cứu.

- Nghiên cứu ảnh vệ tinh và các đặc trưng ảnh vệ tinh.
- Nghiên cứu các phương pháp phát hiện thay đổi rừng, phát hiện mất rừng.
- Nghiên cứu các thuật toán học máy phù hợp để giải bài toán.
- Xây dựng thử nghiệm mô hình và đánh giá hiệu quả trong thực tế.

4. Đối tượng và phạm vi nghiên cứu

❖ Đối tượng nghiên cứu

- Nghiên cứu các đặc trưng của ảnh vệ tinh quang học.
- Nghiên cứu các đặc trưng mạng nơ-ron và học sâu.
- Nghiên cứu thuật toán, mô hình phân phù hợp với giải pháp phát hiện mất rừng.

❖ Phạm vi nghiên cứu

- Thời gian: Từ năm 2017 đến năm 2019.
- Không gian: Phạm vi khu vực đất ven biển có rừng Việt Nam.
- Nguồn dữ liệu sử dụng:
 - Ảnh vệ tinh: Sử dụng ảnh vệ tinh quang học Sentinel-2.
 - Hiện trạng rừng: Sử dụng dữ liệu hiện trạng rừng từ Hệ thống thông tin quản lý ngành Lâm nghiệp - FORMIS.

5. Phương pháp nghiên cứu

Kết hợp nghiên cứu lý thuyết và thực hành:

❖ Lý thuyết: Đọc và tìm hiểu các tài liệu, kiến thức liên quan đến.

- Các đặc trưng ảnh vệ tinh và dữ liệu hiện trạng rừng.
- Các kỹ thuật phát hiện điểm dị thường trong ảnh.
- Các mô hình và thuật toán (mô hình phân lớp, phân vùng).

❖ Thực nghiệm: Cài đặt và đánh giá kết quả thực nghiệm.

- Lựa chọn phương pháp xử lý và thực nghiệm trên bộ dữ liệu huấn luyện.
- Phân tích, đánh giá kết quả tính hiệu quả trong thực tế.

6. Nội dung của luận văn

Luận văn gồm 55 trang khổ A4, 38 hình ảnh, 09 bảng biểu, 02 phụ lục và 15 tài liệu tham khảo. Ngoài phần mở đầu và kết luận, luận văn gồm 4 chương.

Chương 1: Tổng quan vấn đề nghiên cứu.

Trong chương này, luận văn sẽ đi vào tìm hiểu tổng quan vấn đề nghiên cứu về rừng và biến động rừng ven biển; thực trạng và giải pháp. Đặc biệt là tìm hiểu khả năng phát hiện biến động rừng từ ảnh vệ tinh, các phương pháp phát hiện biến động và tính khả thi của việc ứng dụng mạng nơ-ron trong giải bài toán phát hiện biến động mất rừng từ ảnh vệ tinh. Cùng một số công nghệ liên quan khác phục vụ nghiên cứu.

Chương 2: Phát hiện mất rừng ven biển từ ảnh vệ tinh dựa trên mạng nơ-ron.

Trong chương này, nghiên cứu sẽ đi sâu vào việc tìm hiểu mạng nơ-ron tích chập và kiến trúc mạng U-Net để hiểu rõ khả năng ứng dụng, cách thức hoạt động và những ưu điểm nổi bật của kiến trúc, từ đó đi đến định hướng thực tế trong việc giải bài toán phát hiện biến động mất rừng từ ảnh vệ tinh dựa trên mạng nơ-ron. Cuối cùng trong chương này sẽ mô tả cụ thể bài toán và quy trình thực hiện.

Chương 3: Thử nghiệm và đánh giá

Chương này mô tả các cài đặt thử nghiệm và đánh giá kết quả. Quá trình thử nghiệm mô tả rõ: công cụ, môi trường và phạm vi thực hiện; nguồn dữ liệu khai thác, cách thức xây dựng các hàm xử lý, huấn luyện và dự đoán ảnh. Kết quả thử nghiệm mô hình mới sẽ được và so sánh với mô hình đang hoạt động để đưa ra kết quả khách quan.

Chương 4: Kết luận và hướng phát triển

Nội dung chương này tổng kết những kết quả đã đạt được của thử nghiệm, chỉ ra những điểm còn hạn chế và đề xuất hướng phát triển trong tương lai để hoàn thiện.

CHƯƠNG 1

TỔNG QUAN VẤN ĐỀ NGHIÊN CỨU

1.1 Đặt vấn đề

Rừng ven biển (RVB) có vai trò hết sức quan trọng đối với việc phòng hộ và đời sống nhân dân vùng ven biển. Những khu rừng này cung cấp hàng loạt dịch vụ hệ sinh thái lọc nước, ổn định bờ biển, bảo tồn đa dạng sinh học và có vai trò quan trọng đối với ngành du lịch, giải trí, nuôi trồng thủy hải sản đảm bảo sinh kế người dân..vv. Tuy nhiên, hệ sinh thái này lại là một trong những hệ sinh thái dễ bị tổn thương nhất do sự tác động của con người và BĐKH.



Hình 1.1 - Rừng ngập mặn tàn phá sau bão và đê kè biển bị sạt lở do triều cường.

Một trong những nỗ lực giảm thiểu tác động BĐKH trên Toàn cầu và tại Việt Nam trong tương lai đó là tăng cường xây dựng và hỗ trợ hệ sinh thái rừng. Đặc biệt RVB cần được duy trì bền vững, giảm thiểu tối đa các hoạt động xâm lấn hoặc phá rừng.

Để đạt được mục tiêu nêu trên cần liên tục theo dõi các khu vực RVB, theo dõi hoạt động phát triển kinh tế xã hội ven biển để sớm đưa ra quyết định thông minh cho hiện tại và tương lai. Nhiều nhà khoa học đồng ý rằng nên tập trung hệ thống hóa các quy trình khai thác hình ảnh vệ tinh, khai thác cơ sở dữ liệu hiện trạng rừng và đồng thời tối ưu các thuật toán thông minh là giải pháp cốt lõi nâng cao năng lực theo dõi giám sát bảo vệ rừng.

Những năm trở lại đây việc sử dụng hình ảnh chất lượng cao được chụp từ các vệ tinh quan sát Trái đất đang trở thành công nghệ phổ biến để dự đoán và xây dựng bản đồ lớp phủ mặt đất. Các tài liệu nghiên cứu cũng đã chứng minh hiệu suất vượt trội của Mạng nơ-ron tích chập (CNN) cho nhiều nhiệm vụ như: phân loại hình ảnh y tế, xe tự hành, nhận dạng chữ viết... và nhiều lĩnh vực khoa học khác.

Vì vậy, trong nghiên cứu này tôi đề xuất thử nghiệm xây dựng một mạng nơ-ron theo kiến trúc U-Net để huấn luyện học sâu các hình ảnh vệ tinh có độ phân giải cao dựa trên đặc trưng của thực vật và rừng. Hướng đến mục tiêu phân loại nhanh và chính xác ảnh vệ tinh, từ đó có thể tự động dự đoán diện tích rừng, phát hiện biến động rừng ở khu vực ven biển giúp nâng cao năng lực theo dõi giám sát rừng.

1.2 Tổng quan về rừng ven biển và biến động rừng ven biển.

1.2.1 Rừng ven biển.

Rừng là một hệ sinh thái bao gồm các loài thực vật rừng, động vật rừng, nấm, vi sinh vật, đất rừng và các yếu tố môi trường khác, trong đó thành phần chính là một hoặc một số loài cây thân gỗ, tre, nứa, cây họ cau có chiều cao được xác định theo hệ thực vật trên núi đất, núi đá, đất ngập nước, đất cát hoặc hệ thực vật đặc trưng khác; diện tích liên vùng từ 0,3 ha trở lên; độ tàn che từ 0,1 trở lên (Khoản 3, Điều 2, Luật Lâm nghiệp).

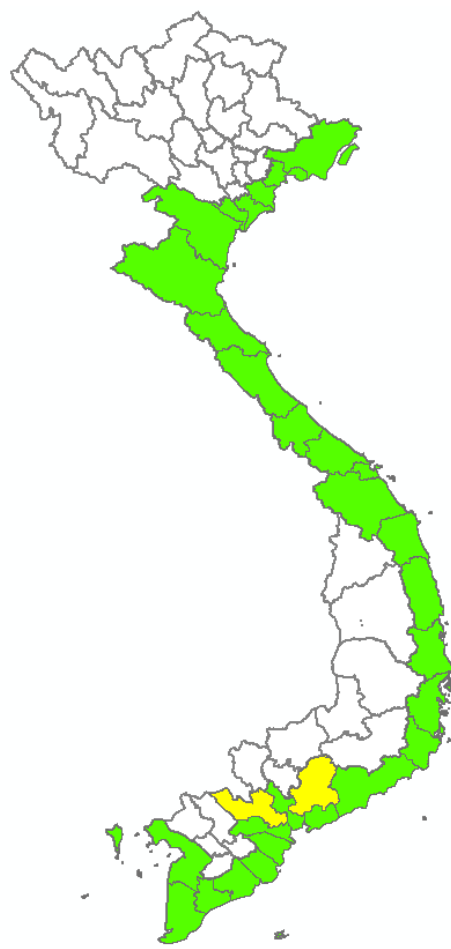
Vùng đất ven biển bao gồm các xã, phường, thị trấn có biển. (Khoản 3, Điều 8, Nghị định 40/2016/NĐ-CP³ ngày 15/05/2016 của Chính phủ: Quy định chi tiết thi hành một số điều của Luật tài nguyên, môi trường biển và hải đảo).

Rừng ven biển bao gồm: Rừng đặc dụng, rừng phòng hộ và đất được quy hoạch để trồng rừng phòng hộ, đặc dụng ở vùng ven biển và hải đảo. (Khoản 2, Điều 1, Nghị định 119/2016/NĐ-CP⁴ ngày 23/8/2016 của Chính phủ: Về một số chính sách quản lý, bảo vệ và phát triển bền vững rừng ven biển ứng phó với biến đổi khí hậu)

Phạm vi rừng ven biển Việt Nam thuộc 600 xã, phường, thị trấn của 130 quận, huyện, thị xã của 28 tỉnh thành phố ven biển (Khoản 1, Điều 1, Quyết định 423/QĐ-TCLN-PTR ngày 31/12/2019 của Tổng cục Lâm nghiệp: Về việc phê duyệt Bộ cơ sở dữ liệu rừng ven biển Việt Nam năm 2018).

Căn cứ trên cơ sở khái niệm về rừng và vùng đất ven biển, cùng các nghị định, quyết định đã ban hành có thể tóm lược ngắn gọn khái niệm về rừng ven biển Việt Nam như sau:

“Rừng ven biển là rừng thuộc các xã, phường, thị trấn có đường bờ biển”



Hình 1.2 - Bản đồ các tỉnh, thành phố có rừng ven biển.

³ <http://datafile.chinhphu.vn/file-remote-v2/DownloadServlet?filePath=vbpq/2016/06/40.signed.pdf>

⁴ <http://datafile.chinhphu.vn/file-remote-v2/DownloadServlet?filePath=vbpq/2016/08/119.signed.pdf>

1.2.2 Hiện trạng rừng ven biển Việt Nam.

Theo Quyết định 423/QĐ-TCLN-PTR ngày 31/12/2019, tổng diện tích rừng và đất chưa có RVB Việt Nam năm 2018 là: 709.013 ha. Trong đó diện tích có rừng 454.337 ha.

Theo thống kê báo cáo⁵ từ bộ CSDL RVB, diện tích có rừng (gồm rừng tự nhiên và rừng trồng) năm 2017 là: 444.041 ha [12]. Điều đó cho thấy diện tích có rừng sau một năm đã tăng lên 10,295 ha.

Tuy nhiên, từ số liệu chi tiết thấy rằng:

- Diện tích rừng tăng: Phần lớn là do rừng trồng đạt tuổi thành rừng (tăng 10,203 ha), còn rừng tự nhiên tăng không đáng kể (tăng 93 ha).
- Diện tích rừng giảm: Giảm nhiều nhất, chính là rừng tự nhiên trên núi đá (giảm 252 ha) và RNM (giảm 789 ha).



Hình 1.3 – Bản đồ hiện trạng RVB Việt Nam (31/12/2018)

TT	Phân loại rừng	Diện tích RVB 2017	Diện tích RVB 2018	Diện tích tăng/giảm
A	DIỆN TÍCH CÓ RỪNG	444.041,30 ha	454,337,15 ha	Tăng 10.296 ha
I	Rừng tự nhiên	215,173.01 ha	215,266.00 ha	Tăng 93 ha
1	Trên núi đất	137,981.16 ha	138,904.86 ha	Tăng 924 ha
2	Trên núi đá	27,467.33 ha	27,214.91 ha	Giảm 252 ha
3	Trên đất ngập nước	49,724.52 ha	49,146.23 ha	Giảm 578 ha
	- Ngập mặn	46,997.28 ha	46,208.64 ha	Giảm 789 ha
	- Trên đất phèn	2,687.52 ha	2,897.87 ha	Tăng 210 ha
	- Ngập nước ngọt	39.72 ha	39.72 ha	-
4	Trên cát	-	-	-

⁵ <http://rungvenbien.ifee.edu.vn/thongkebaocao>

TT	Phân loại rừng	Diện tích RVB 2017	Diện tích RVB 2018	Diện tích tăng/giảm
II	Rừng trồng	228,868.29 ha	239,071.15 ha	Tăng 10,203 ha
1	Trên núi đất	124,849.28 ha	127,724.79 ha	Tăng 2,876 ha
2	Trên núi đá	1,714.95 ha	1,704.80 ha	Giảm 10 ha
3	Trên đất ngập nước	72,977.48 ha	79,439.48 ha	Tăng 6,462 ha
	- Ngập mặn	68,091.24 ha	70,865.08 ha	Tăng 2,774 ha
	- Trên đất phèn	4,167.89 ha	7,923.59 ha	Tăng 3,756 ha
	- Ngập nước ngọt	718.35 ha	650.81 ha	Giảm 68 ha
4	Trên cát	29,326.58 ha	30,202.08 ha	Tăng 876 ha

Bảng 1.1 - Số sánh diện tích RVB có rừng năm 2017 và năm 2018.

Từ hiện trạng RVB công bố có thể thấy rằng những năm qua chủ chương của nhà nước trong công tác trồng phục hồi RVB đã được các địa phương đã thực hiện tốt, tuy nhiên một diện tích không nhỏ rừng tự nhiên ven biển vẫn bị suy giảm bởi nhiều nguyên nhân khác nhau.

1.2.3 Nguyên nhân biến động rừng ven biển.

❖ Khái niệm chung về biến động.

Cụm từ biến động được hiểu là sự biến đổi, thay đổi, thay thế trạng thái này bằng một trạng thái khác liên tục của sự vật, hiện tượng tồn tại trong môi trường tự nhiên cũng như môi trường xã hội.

Phát hiện biến động là quá trình nhận dạng sự khác biệt về trạng thái của một đối tượng hay hiện tượng bằng cách quan sát chúng tại những thời điểm khác nhau [1].

❖ Biến động rừng & biến động rừng ven biển.

Biến động rừng là việc đánh giá sự thay đổi tăng hoặc giảm về diện tích rừng trong một giai đoạn nhất định nào đó.

Trên cơ sở khái niệm chung biến động, khái niệm rừng ven biển và hiểu biết về việc xác định biến động rừng, ta có thể khái quát nhanh biến động rừng ven biển: *Biến động rừng ven biển là giá sự thay đổi về diện tích rừng thuộc các xã, phường, thị trấn có đường bờ biển trong một khoảng thời gian nhất định.*

❖ Nguyên nhân biến động rừng ven biển Việt Nam.

Để theo dõi, phân tích đánh giá biến động RVB phải nắm được các nguyên nhân cơ bản dẫn đến biến động. Một số nguyên nhân biến động diện tích RVB:

▪ *Biến động mất rừng (giảm rừng)*

- Thiếu quy hoạch các dự án nuôi trồng thủy sản.
- Quá trình đô thị hoá phát triển kinh tế ven biển.
- Hoạt động xây dựng cầu cảng, công trình lấn biển.
- Ảnh hưởng của biến đổi khí hậu, thiên tai bão lũ, sạt lở..vv

▪ *Biến động tăng rừng*

- Diện tích rừng tăng chủ yếu là nhờ rừng được trồng phục hồi và một phần được tái sinh tự nhiên.

1.3 Các nghiên cứu liên quan phát hiện biến động rừng từ ảnh vệ tinh.

Theo dõi, cập nhật diễn biến rừng, dựa trên nền kết quả kiểm kê rừng, là công việc hết sức cần thiết để đảm bảo giá trị lâu dài của cơ sở dữ liệu. Phát hiện vị trí mất rừng luôn là công việc khó khăn và mất nhiều công sức [2].

Trong nhiều năm qua, ứng dụng viễn thám để phân loại đánh giá trạng thái lớp phủ mặt đất đã được sử dụng rộng rãi trong nhiều lĩnh vực, trong đó có giám sát biến động diện tích rừng ở nhiều quy mô khác nhau.

1.3.1 Khả năng khai thác ảnh vệ tinh phục vụ phát hiện biến động rừng.

Việc kết hợp các loại tư liệu vệ tinh quang học khác nhau sẽ làm tăng cơ hội cho việc có được ảnh không mây phục vụ thường xuyên hơn cho mục tiêu giám sát bề mặt trái đất. Hiện nay, các vệ tinh quang học như: Landsat-8 OLI (từ 2013), Sentinel-2 MSI (Sentinel-2A - từ 2015; Sentinel-2B - từ 2017) hiện đang cung cấp ảnh miễn phí trên phạm vi toàn cầu với độ phân giải không gian từ 10 đến 30 m tạo ra cơ hội rất lớn để giám sát bề mặt trái đất [3].

Nếu không xét đến ảnh hưởng của mây và bóng mây thì trung bình tại một vị trí tại nước ta, chu kỳ lặp lại của ảnh: 15 ngày (Landsat-8), 5 ngày (Sentinel-2) và 3 ngày (kết hợp hai loại ảnh). Nhưng nếu loại bỏ ảnh hưởng của mây và bóng mây thì, tính trung bình cho mọi vị trí tại Việt Nam, chu kỳ lặp lại của ảnh: 30 ngày (Landsat-8), 10 ngày (Sentinel-2) và 7 ngày (kết hợp hai loại ảnh). Tương ứng với tỷ lệ số ảnh Landsat-8 và Sentinel-2 có thể sử dụng cho mục tiêu giám sát bề mặt so với tổng số ảnh thu được tương ứng là 51% và 39% [3].

Với chu kỳ lặp lại của ảnh như vậy, tư liệu ảnh Landsat-8 OLI phù hợp cho mục tiêu giám sát biến động hàng năm. Trong khi đó, tư liệu ảnh Sentinel-2 MSI phù hợp cho mục tiêu giám sát biến động hàng quý. Sự kết hợp của cả hai loại tư liệu ảnh có thể đáp ứng được mục tiêu giám sát biến động trên mặt đất hàng tháng [3].

Ngoài ra, sự kết hợp giữa ảnh vệ tinh quang học Sentinel-2 và ảnh vệ tinh radar Sentinel-1 cho khả năng phát hiện mất rừng, kết quả thu được tốt hơn so với từng loại ảnh riêng lẻ [4].

1.3.2 Một số kết quả nghiên cứu về phát hiện biến động rừng từ ảnh vệ tinh.

Sử dụng tư liệu ảnh vệ tinh là hướng đi nghiên cứu góp phần mang lại sự biến đổi mạnh mẽ trong hoạt động quản lý các nguồn tài nguyên thiên nhiên nói chung và tài nguyên rừng nói riêng. Trên thế giới và ở Việt Nam đã có rất nhiều công trình nghiên cứu ứng dụng ảnh vệ tinh, công nghệ viễn thám (RS) và hệ thống thông tin địa lý (GIS) phục vụ cho công tác quản lý tài nguyên và môi trường. Có thể kể đến các công trình nghiên cứu tiêu biểu:

- Giám sát biến động RNM Cần Giờ TP. HCM sử dụng kỹ thuật viễn thám và GIS [1]. Biến động RNM theo thời gian có thể được giám sát bằng cách sử dụng phương pháp phân tích biến động sau phân loại. Trước tiên dữ liệu ảnh vệ tinh đa phổ từng thời điểm được tiến hành phân loại độc lập. Sau đó sử dụng GIS để tiến hành phát hiện biến động bằng cách so sánh ảnh phân loại của cùng một vùng tại hai thời điểm khác nhau. Kết quả cho thấy sử dụng dữ liệu ảnh vệ tinh Landsat và Aster để thành lập bản đồ RNM đã đạt độ chính xác xấp xỉ gần 80%.

- Ứng dụng công nghệ tích hợp tư liệu viễn thám và hệ thống tin địa lý xác định sự biến động rừng ngập mặn [5]. Nghiên cứu sử dụng tư liệu ảnh SPOT độ phân giải 20 m, dùng phương pháp phân loại kết hợp giữ chỉ số thực vật (NDVI), thành phần tỉ số nhiều tối thiểu (MNF) và phân loại theo thuật toán xác xuất cực đại (MLC) để thành lập bản đồ biến động RNM.

- Xác định vị trí mất rừng bằng phương pháp phân tích véc tơ thay đổi đa biến (MCVA) trên tư liệu vệ tinh Landsat-8 [2]. Nghiên cứu này là sử dụng phương pháp phân tích véc tơ thay đổi đa biến (Multi-variant Change Vector Analysis - MCVA) dựa trên tư liệu ảnh viễn thám Landsat-8. Với hai véc tơ đầu vào: (1) chỉ số thực vật khác biệt chuẩn và (2) chỉ số đất khác biệt chuẩn. Kết quả nghiên cứu cho thấy có thể phát hiện vị trí mất rừng từ ảnh vệ tinh phục vụ cho công tác quản lý, theo dõi và cập nhật diễn biến rừng.

- Nghiên cứu kết hợp ảnh vệ tinh quang học Sentinel-2 và Radar Sentinel-1 trong phát hiện mất rừng ở tỉnh Gia Lai [4]. Bằng việc kết hợp chỉ số NDVI của ảnh vệ tinh quang học Sentinel-2 với giá trị tán xạ ngược của ảnh radar Sentinel-1 để phát hiện mất rừng trên địa bàn tỉnh Gia Lai giữa 2 thời kỳ khác nhau. Kết quả chỉ ra rằng sự kết hợp giữa ảnh vệ tinh quang học Sentinel-2 và ảnh vệ tinh radar Sentinel-1 cho khả năng phát hiện mất rừng ở Gia Lai tốt hơn so với dùng từng loại riêng lẻ.

Và một số nghiên cứu khác:

Sử dụng ảnh Google Earth để xây dựng bản đồ hiện trạng rừng và đánh giá biến động rừng tại Công ty Lâm nghiệp La Ngà, Tỉnh Đồng Nai [6].

Ứng dụng GIS và ảnh Landsat đa thời gian xây dựng bản đồ biến động diện tích rừng tại xã vùng đệm Xuân Đài và Kim Thượng, Vườn quốc gia Xuân Sơn [7].

Phương pháp tích hợp dữ liệu MODIS và Landsat để giám sát có hệ thống về độ che phủ và thay đổi rừng trong Lưu vực Congo [9].

Giám sát rừng ngập mặn trong Google Earth Engine [10].

1.3.3 Khó khăn và thách thức trong phát hiện biến động rừng từ ảnh vệ tinh.

RVB có đặc điểm mọc tự nhiên hoặc được trồng và sinh trưởng ở những vùng đất, khu vực tiếp giáp giữa đất liền và biển. Do đó, có bốn đối tượng chính thể hiện trên tư liệu ảnh viễn thám quang học là: *thảm thực vật, đất, nước và mây*. Riêng với RVB còn bị ảnh hưởng bởi các đợt thủy triều trong ngày và theo mùa. Những yếu tố này tác động lớn tới đặc trưng quang học của ảnh.



Hình 1.4 - Ảnh chụp bề mặt trái đất từ vệ tinh Sentinel-2 ngày 04/06/2020

Ngoài ra, rừng ven biển gồm nhiều trạng thái và loại rừng khác nhau như: rừng thông chắn cát, rừng tự nhiên trên núi ven biển, rừng ngập mặn..vv điều này làm gia tăng khó khăn trong việc xác định biến động rừng ven biển.

Đã có nhiều phương pháp phát hiện biến động lớp phủ sử dụng dữ liệu số đã được đề xuất như: so sánh các phân loại lớp phủ; phân loại ảnh đa thời gian; ảnh hiệu hoặc ảnh chia; sự khác biệt về chỉ số thực phủ; phân tích thành phần chính [1]. Tuy nhiên, kết quả thực hiện từ các nghiên cứu đã công bố cho thấy các phương pháp trên không có phương pháp nào thực sự vượt trội.

Với phương pháp so sánh sau phân loại và phương pháp phân loại đa thời gian trực tiếp đều phải xác định ngưỡng phân chia bằng thực nghiệm để tách các điểm ảnh biến động và không biến động. Trong thực tế, việc xác định ngưỡng phân chia chính xác không là điều dễ dàng. Phương pháp MCVA [2] là một trường hợp như vậy, tác giả cần đến hai ngưỡng chỉ là ngưỡng chỉ số thực vật và ngưỡng chỉ số mất rừng. Các ngưỡng chỉ số chỉ phù hợp ở vùng nhất định và cần thử nghiệm giá trị mới nếu thay đổi hoặc mở rộng phạm vi áp dụng.

Hay phương pháp phân loại sử dụng tập các kênh đa phổ ở các thời điểm khác nhau để phát hiện biến động đòi hỏi phải chọn được các vùng mẫu thể hiện các vùng biến động và không biến động một cách phù hợp. Và điều này cũng không là vấn đề đơn giản đối với người ít chuyên môn, hiểu biết về ảnh vệ tinh.

Cũng cần nói thêm rằng có một số phương pháp chính xác hơn như: sử dụng máy bay không người lái (UAV) hoặc tuần tra đo đạc địa hình. Nhưng phương pháp này thường rất tốn kém, cần nhân sự có trình độ chuyên môn cao, khó triển khai mở rộng liên tục hoặc giám sát tự động trong phạm vi lớn. Do nhiều vị trí địa hình phức tạp con người không thể đi đến và tiếp cận ngay được.

1.4 Ứng dụng mạng nơ-ron giải bài toán phát hiện mất rừng từ ảnh vệ tinh.

Những năm qua, với sự bùng nổ mạnh mẽ của cuộc cách mạng công nghiệp 4.0, các thuật ngữ như trí tuệ nhân tạo (Artificial Intelligence - AI), học máy (Machine Learning - ML), học sâu (Deep Learning - DL), mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) đang dần trở nên phổ biến và quen thuộc được nhắc đến thường xuyên như một bước đột phá của kỷ nguyên công nghệ 4.0.

Đặc biệt, mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) là một trong những mô hình DL phổ biến, được ứng dụng nhiều trong giải quyết các bài toán như nhận dạng ảnh, phân tích video, xử lý ngôn ngữ tự nhiên và hầu hết đều giải quyết tốt các bài toán này; một số ứng dụng thực tế như: phát hiện bệnh bằng hình ảnh y tế, robot phẫu thuật, ô tô tự lái, hệ thống dịch tự động...vv. Điều đó cho thấy tiềm năng to lớn của ứng dụng CNN trong giải bài toán phát hiện biến động rừng từ ảnh vệ tinh và khắc phục hạn chế của các phương pháp cũ trước đây.

Trên thế giới đã nhiều nhà khoa học tiếp cận, nhiều công trình nghiên cứu ứng dụng DL, CNN để giải bài toán phát hiện biến động rừng. Các nghiên cứu tiêu biểu như: Phân loại thời tiết, địa hình và nạn phá rừng của Amazon bằng cách sử dụng CNN đa nhiệm sâu [11]; Sản xuất bản đồ che phủ mặt đất với độ phân giải cao từ chuỗi ảnh và CNN [12]; Phát hiện biến động mất rừng tại Brazil sử dụng dữ liệu Landsat và CNN [13]; Phát hiện thay đổi rừng trong ảnh vệ tinh không đầy đủ với DNN [14].

Các nghiên cứu đều cho thấy ứng dụng NN nói chung và CNN nói riêng cho phép phân tích xử lý giải quyết bài toán phát hiện đối tượng từ ảnh vệ tinh nhanh, hiệu quả, có khả năng mở rộng và cải thiện độ chính xác hơn nữa nhờ khả năng học hỏi.

1.5 Một số công nghệ liên quan.

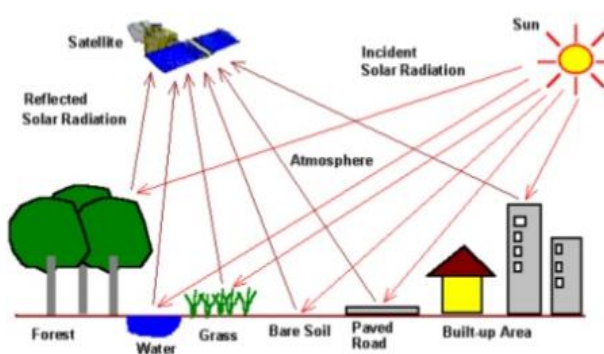
1.5.1 Viễn thám.

Viễn thám (Remote Sensing - RS) được hiểu là lĩnh vực khoa học thu nhận thông tin về một đối tượng, một khu vực hoặc một hiện tượng thông qua việc phân tích tư liệu thu nhận được bằng các phương tiện. Những phương tiện này không có sự tiếp xúc trực tiếp với đối tượng, khu vực hoặc với hiện tượng được nghiên cứu. Thực hiện được những công việc đó chính là thực hiện viễn thám.

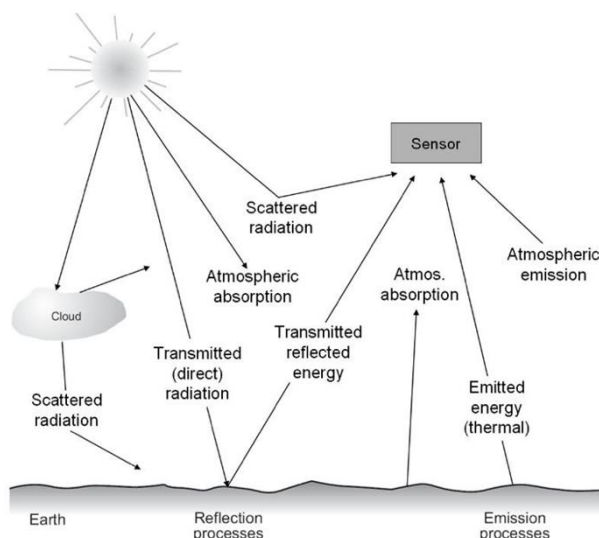
Mặc dù có rất nhiều định nghĩa khác nhau về viễn thám, nhưng mọi định nghĩa đều có nét chung, nhấn mạnh *“Viễn thám là lĩnh vực khoa học công nghệ cho phép nghiên cứu, thu nhận từ xa các thông tin về đối tượng, hiện tượng trên bề mặt Trái Đất mà không cần tiếp xúc trực tiếp với chúng”*.

Cơ sở khoa học của công nghệ viễn thám dựa trên bản chất vật lý trong tự nhiên là các vật thể (đối tượng) trong những điều kiện khác nhau thì khả năng phản xạ hoặc bức xạ của sóng điện từ sẽ có những đặc trưng riêng.

Nguồn năng lượng mặt trời (sóng điện từ) chiếu tới mặt đất, năng lượng của nó sẽ tác động lên bề mặt Trái Đất. Nguồn năng lượng mặt trời gồm 3 phần: một phần năng lượng tới của bức xạ bị hấp thụ, một phần được truyền qua sau đó phản xạ trở lại và một phần bị phản xạ ngay từ bề mặt của đối tượng [5].



Tỷ lệ giữa ba phần năng lượng này khác nhau khi bức xạ truyền tới các đối tượng tự nhiên khác nhau. Như vậy, chính nhờ sự khác biệt về tỷ lệ giữa các phần năng lượng đó là cơ sở để phân biệt các đối tượng tự nhiên [5].



Hình 1.5 - Tương tác năng lượng trong khí quyển và trên bề mặt Trái Đất

Để nghiên cứu sự phụ thuộc của năng lượng phản xạ phổ vào bước sóng người ta đưa ra khái niệm phản xạ phổ theo phần trăm (%) bằng công thức sau:

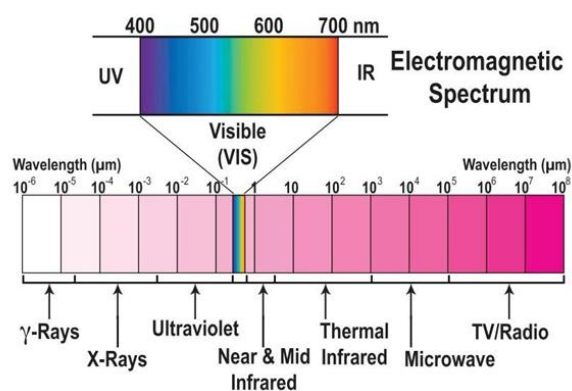
$$R(\lambda) = E_{px}(\lambda) / E_0(\lambda) \cdot 100\% \quad (2.3)$$

Trong đó: $R(\lambda)$: phản xạ phổ theo phần trăm (%)

$E_{px}(\lambda)$: năng lượng phản xạ

(E_0) : năng lượng mặt trời chiếu xuống mặt đất

Các đối tượng tự nhiên bao gồm tất cả các đối tượng thuộc lớp phủ bề mặt Trái Đất, các đối tượng tự nhiên trên mặt đất rất đa dạng và phức tạp. Đặc trưng phản xạ phổ của các đối tượng tự nhiên phụ thuộc vào nhiều yếu tố như điều kiện chiếu sáng, môi trường, khí quyển và bề mặt đối tượng cũng như bản chất của đối tượng đó.

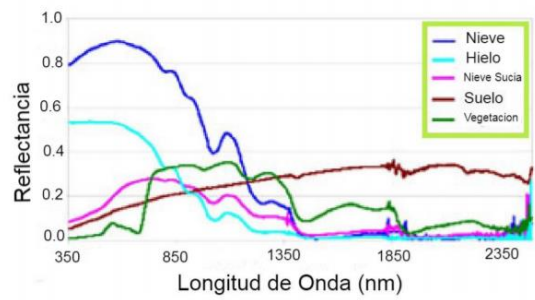


Hình 1.6 - Các dải phổ dùng trong viễn thám.

Tất cả các vật thể đều phản xạ, hấp thụ, phân tích và bức xạ sóng điện từ bằng các cách thức khác nhau và các đặc trưng này thường được gọi là đặc trưng phổ. Đặc trưng phổ sẽ được phân tích theo nhiều cách khác nhau để nhận dạng ra đối tượng trên bề mặt đất, nó sẽ cho phép giải thích được mối quan hệ giữa đặc trưng phổ và sắc, tông màu trên ảnh tổ hợp màu để giải đoán đối tượng.

Đặc trưng phản xạ phổ của nhóm đối tượng phụ thuộc vào các bước sóng và thường chia làm 3 nhóm đối tượng chính:

- Nhóm 1: Lớp phủ thực vật.
- Nhóm 2: Lớp đối tượng đất.
- Nhóm 3: Lớp đối tượng nước.



Hình 1.7 - Phân xạ phổ tương ứng với các loại đối tượng

1.5.2 Ảnh vệ tinh.

Ảnh vệ tinh (ảnh viễn thám): là ảnh số thể hiện các vật thể trên bề mặt trái đất được thu nhận bởi các bộ cảm biến đặt trên vệ tinh. Ảnh vệ tinh có thể được lưu theo các kênh ảnh đơn (trắng đen) ở dạng số trong máy tính hoặc các kênh ảnh được tổ hợp (ảnh màu) tùy theo mục đích người sử dụng.

❖ Phân loại ảnh vệ tinh

- Ảnh quang học: là loại ảnh được tạo ra bởi việc thu nhận các bước sóng ánh sáng nhìn thấy (bước sóng 0.4 - 0.76 micromet).
- Ảnh hồng ngoại: là loại ảnh được tạo ra bởi việc thu nhận các bước sóng hồng ngoại phát ra từ vật thể (bước sóng 8 - 14 micromet).
- Ảnh radar: là loại ảnh được tạo ra bởi việc thu nhận các bước sóng trong dải sóng siêu cao tần (bước sóng lớn hơn 2 cm).

❖ Phạm vi ứng dụng của ảnh viễn thám



Hình 1.8 - Ứng dụng viễn thám trong quản lý đô thị.

- **Khí tượng:** Dùng để dự báo thời tiết, dự báo thiên tai liên quan đến biến đổi khí hậu bề mặt đất, mây, mặt nước...vv
- **Bản đồ:** Công cụ đắc lực phục vụ cho ngành bản đồ, thành lập các loại bản đồ địa hình và bản đồ chuyên đề ở nhiều tỉ lệ khác nhau.
- **Nông Lâm nghiệp:** theo dõi biến đổi thảm thực vật, độ che phủ rừng..vv
- **Địa chất:** Theo dõi tốc độ sa mạc hoá, tốc độ xâm thực bờ biển, phân tích những cấu trúc địa chất trên mặt cũng như bên trong lòng đất..vv
- **Môi trường:** Giám sát biến động ô nhiễm, rò rỉ dầu trên mặt, nghiên cứu quản lý biến động đô thị hóa, nghiên cứu hiện tượng đảo nhiệt đô thị..vv

1.5.2.1 Ảnh vệ tinh Sentinel-2.

Sentinel-2 là một nhiệm vụ quan sát Trái Đất từ chương trình Copernicus⁶ có hệ thống thu được hình ảnh quang học độ phân giải cao, đa quang phổ trên đất liền và vùng nước ven biển. Sentinel-2 bao gồm 2 vệ tinh Sentinel-2A (phóng lên không gian ngày 23/06/2015) và Sentinel-2B (phóng lên không gian ngày 07/07/2017).

Nhiệm vụ Sentinel-2 cung cấp dữ liệu ảnh có tầm nhìn rộng 290 km, gồm 13 dải quang phổ⁷ (tương ứng với 13 kênh ảnh) với độ phân giải 10 m, 20m, 60 m. Chu kỳ chụp của mỗi vệ tinh tại một vị trí là 10 ngày và 5 ngày khi kết hợp hai vệ tinh.

Bảng 1.2 - Danh sách các kênh ảnh, bước sóng và độ phân giải của ảnh Sentinel-2⁸

Band Number	S2A		S2B		Spatial resolution (m)
	Central wavelength (nm)	Bandwidth (nm)	Central wavelength (nm)	Bandwidth (nm)	
1	442.7	21	442.3	21	60
2	492.4	66	492.1	66	10
3	559.8	36	559.0	36	10
4	664.6	31	665.0	31	10
5	704.1	15	703.8	16	20
6	740.5	15	739.1	15	20
7	782.8	20	779.7	20	20
8	832.8	106	833.0	106	10
8a	864.7	21	864.0	22	20
9	945.1	20	943.2	21	60
10	1373.5	31	1376.9	30	60
11	1613.7	91	1610.4	94	20
12	2202.4	175	2185.7	185	20

⁶ Copernicus là một hệ thống giám sát Trái đất của châu Âu. Dữ liệu được thu thập bởi các nguồn khác nhau, bao gồm các vệ tinh quan sát Trái đất và cảm biến tại chỗ. Dữ liệu được xử lý và cung cấp thông tin đáng tin cậy và cập nhật trong sáu lĩnh vực chủ đề: đất, biển, khí quyển, biến đổi khí hậu, quản lý khẩn cấp và an ninh.

⁷ <https://earth.esa.int/web/sentinel/technical-guides/sentinel-2-msi/msi-instrument>,

http://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-2/Facts_and_figures

⁸ <https://earth.esa.int/web/sentinel/technical-guides/sentinel-2-msi/msi-instrument>

Ảnh từ vệ tinh Sentinel-2 (sau đây gọi tắt là ảnh Sentinel-2) gồm các sản phẩm dữ liệu Level-1B, Level-1C and Level-2A. Trong đó, dữ liệu Level-2A tạo ra, từ các thuật toán phân loại cảnh và hiệu chỉnh khí quyển.

Ảnh Sentinel-2 hỗ trợ các nghiên cứu giám sát mặt đất, bao gồm: giám sát thảm thực vật (nông nghiệp, rừng), đất và nước; giám sát các vùng thủy nội địa và các khu vực ven biển...vv.



Hình 1.9 - Vệ tinh Sentinel-2 chụp ảnh bề mặt Trái Đất.

1.5.2.2 Chỉ số quang phổ đặc trưng⁹.

Các chỉ số quang phổ là sự kết hợp tham số đơn giản của các kênh quang phổ khác nhau, chúng được thiết kế để tăng cường độ nhạy cảm với các hiện tượng sinh lý cụ thể, chẳng hạn như cây xanh của thảm thực vật, hàm lượng nước.

❖ Chỉ số thực vật khác biệt chuẩn - NDVI

Chỉ số thực vật khác biệt chuẩn (Normalized Difference Vegetation Index - NDVI) là một thuật toán tiêu chuẩn được thiết kế để đo chất lượng thảm thực vật trên mặt đất, dựa trên cách cây phản xạ ánh sáng.

Công thức tính giá trị NDVI cho ảnh Sentinel-2:

$$\text{Normalized Difference Vegetation Index (NDVI)} = \frac{\text{Band 8} - \text{Band 4}}{\text{Band 8} + \text{Band 4}}$$

Trong đó:

- Band 8: là giá trị phản xạ phổ của kênh cận hồng ngoại gần (nir).
- Band 4: là giá trị phản xạ phổ của kênh đỏ (red).

⁹ <https://earth.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-2a/algorithm>

Chỉ số NDVI xác định các giá trị từ -1,0 đến 1,0; về cơ bản đây là chỉ số là đại diện cho màu xanh lá cây, trong đó các giá trị âm chủ yếu được hình thành từ mây, nước và tuyết và các giá trị gần bằng 0 chủ yếu được hình thành từ đá và đất trống. Các giá trị rất nhỏ (0,1 hoặc ít hơn) của hàm NDVI tương ứng với các khu vực trống của đá, cát hoặc tuyết. Giá trị vừa phải (từ 0,2 đến 0,3) đại diện cho cây bụi và đồng cỏ, trong khi giá trị lớn (từ 0,6 đến 0,8) cho thấy rừng ôn đới và nhiệt đới.

Giá trị NDVI ¹⁰	Lớp phủ bề mặt đất
< 0,1	Khu vực trống, bê tông, đá, cát, mặt nước hoặc tuyết
0,1 – 0,2	Đất đá cằn cỗi, cây bụi
0,2 – 0,3	Cây bụi, đồng cỏ và đất nông nghiệp dễ trồng
0,3 – 0,6	Đồng cỏ, cây trồng nông nghiệp, rừng thưa
> 0,6	Rừng nhiệt đới

Bảng 1.3 - Phân loại giá trị NDVI theo chất lượng thực vật trong lớp phủ bề mặt trái đất

❖ Chỉ số tuyết khác biệt chuẩn - NDSI

Chỉ số tuyết khác biệt chuẩn (Normalised Difference Snow Index – NDSI) là thước đo độ chênh lệch phản xạ giữa kênh xanh lục và kênh hồng ngoại sóng ngắn.

Công thức tính giá trị NDSI cho ảnh Sentinel-2:

$$\text{Normalised Difference Snow Index (NDSI)} = \frac{\text{Band 3} - \text{Band 11}}{\text{Band 3} + \text{Band 11}}$$

Trong đó

- Band 3: là giá trị phản xạ phổ của kênh xanh lục (green)
- Band 11: là giá trị phản xạ phổ của kênh hồng ngoại sóng ngắn (swir)

Chỉ số NDSI hữu ích trong việc phân biệt giữa mây và tuyết, là một trong những chỉ số cần thiết được sử dụng trong phương pháp loại bỏ mây được trình bày ở phần sau.

¹⁰ <https://www.agricolus.com/en/indici-vegetazione-ndvi-ndmi-istruzioni-luso/#ndvi>

❖ Chỉ số nước khác biệt chuẩn – NDWI

Chỉ số nước khác biệt bình chuẩn (Normalized Difference Water Index - NDWI), được giới thiệu lần đầu tiên vào năm 1996 bởi Gao, phản ánh độ ẩm trong cây và đất. NDWI giúp có thể nhận ra ngay các khu vực đất trồng có vấn đề về ngập nước.

Công thức tính giá trị NDWI cho ảnh Sentinel-2:

$$\text{Normalized Difference Water Index (NDWI)} = \frac{\text{Band 8} - \text{Band 11}}{\text{Band 8} + \text{Band 11}}$$

Trong đó

- *Band 8*: là giá trị phản xạ phổ của kênh cận hồng ngoại gần (*nir*)
- *Band 11*: là giá trị phản xạ phổ của kênh hồng ngoại sóng ngắn (*swir*)

NDWI xác định các giá trị từ -1,0 đến 1,0

Giá trị NDWI ¹¹	Lớp phủ bề mặt đất
< 0	Độ tàn che thấp, ẩm ướt và ngập nước
0,1 – 0,4	Độ tàn che trung bình, ngập thấp
0,4 – 0,6	Độ tàn che cao, không bị ngập nước

Bảng 1.4 - Bảng phân loại NDWI theo chất lượng thực vật trong lớp phủ bề mặt trái đất.

1.5.3 Hệ thống thông tin địa lý.

Hệ thống thông tin địa lý (Geographic Information System - GIS) là một hệ thống máy tính giúp thu thập, xử lý, chỉnh sửa, lưu trữ, quản lý, chia sẻ, phân tích, mô hình hóa và trực quan hóa khối lượng dữ liệu lớn để hiểu mối quan hệ không gian, mô hình, xu hướng và đưa ra quyết định. Một hệ thống GIS cho phép quản lý, phân tích dữ liệu không gian và tạo kết quả thể hiện dưới dạng thích hợp như: văn bản, đồ họa, bảng biểu..vv. GIS được sử dụng nhằm xử lý đồng bộ các lớp thông tin không gian gắn với thông tin thuộc tính, phục vụ nghiên cứu, quy hoạch và quản lý các hoạt động theo lãnh thổ.

Có nhiều định nghĩa về GIS, nhưng nói chung đã thống nhất quan niệm chung: GIS là một hệ thống kết hợp giữa con người và hệ thống máy tính cùng các thiết bị ngoại vi để lưu trữ, xử lý, phân tích, hiển thị các thông tin địa lý để phục vụ một mục đích nghiên cứu, quản lý nhất định.

¹¹ <https://www.agricolus.com/en/indici-vegetazione-ndvi-ndmi-istruzioni-luso/#ndwi>

❖ Chức năng cơ bản của GIS:

- **Thu thập dữ liệu:** GIS cung cấp công cụ để tích hợp dữ liệu từ nhiều nguồn khác nhau thành một định dạng chung để so sánh và phân tích.
- **Quản lý dữ liệu:** Sau khi sử dụng dữ liệu thu thập và tích hợp, GIS cung cấp các chức năng lưu trữ và duy trì dữ liệu.
- **Phân tích dữ liệu:** Là chức năng quan trọng nhất của GIS nó cung cấp các chức năng như nội suy không gian, tạo vùng đệm, chồng lớp.
- **Hiển thị kết quả:** GIS cho phép hiển thị dữ liệu trực quan với nhiều kiểu khác nhau như: dạng bảng, biểu đồ, bản đồ... Vì vậy, người sử dụng có thể thông qua GIS tương tác hiệu quả với dữ liệu.

❖ GIS trong nghiên cứu biến động rừng.

Ảnh viễn thám sau khi phân tích sẽ thể hiện sự phân bố của các đối tượng theo không gian và thời gian. Kết quả xử lý ảnh viễn thám sẽ chỉ ra hiện trạng lớp phủ rừng tại thời điểm chụp ảnh. Bằng việc chồng xếp và phân tích, GIS cho phép tích hợp từ các kết quả phân loại của nhiều thời điểm để thành lập nhanh và chính xác bản đồ rừng.

Với chức năng cung cấp hiển thị thông tin thay đổi giữa các loại hình lớp phủ theo từng thời điểm yêu cầu hoặc thao đơn vị hành chính, GIS có thể cho phép người sử dụng giám sát quá trình biến động rừng theo từng loại hình và các khoảng thời gian khác nhau.

1.5.4 Điện toán đám mây

Điện toán đám mây (Cloud Computing) là mô hình điện toán sử dụng các công nghệ máy tính và phát triển dựa vào mạng Internet. Điện toán đám mây cung cấp linh hoạt các tài nguyên máy tính cho người dùng tùy theo mục đích sử dụng thông qua kết nối Internet.

Với đám mây, có thể triển khai các dịch vụ công nghệ một cách nhanh chóng, mở rộng sang các khu vực địa lý mới hoặc triển khai trên toàn cầu trong vài phút.

1.5.4.1 Google Earth Engine.

Google Earth Engine (GEE) là một nền tảng dựa trên đám mây để phân tích không gian địa lý trên quy mô toàn cầu. GEE mang đến khả năng tính toán khổng lồ của Google để giải quyết nhiều vấn đề xã hội bao gồm phá rừng, hạn hán, thảm họa, bệnh tật, an ninh lương thực, quản lý nước, giám sát khí hậu và bảo vệ môi trường. GEE là duy nhất trong lĩnh vực này, nó là một nền tảng tích hợp (sử dụng siêu máy tính và tài nguyên điện toán đám mây quy mô lớn) được thiết kế cung cấp năng lực tính toán không chỉ dành riêng cho các nhà khoa học viễn thám, mà còn rộng lớn hơn thế [8].

Earth Engine bao gồm một danh mục dữ liệu sẵn sàng phân tích (petabyte) cùng với một dịch vụ tính toán song song hiệu năng cao. Nó được truy cập và kiểm soát thông qua giao diện lập trình ứng dụng (API) có thể truy cập Internet và môi trường phát triển tương tác dựa trên web cho phép tạo mẫu nhanh và trực quan hóa kết quả [8].

❖ **Các thành phần chính của Earth Engine:**

- Bộ dữ liệu (Datasets): Kho lưu trữ lớn với quy mô hàng petabyte dữ liệu hình ảnh viễn thám được công bố rộng rãi và các dữ liệu khác.

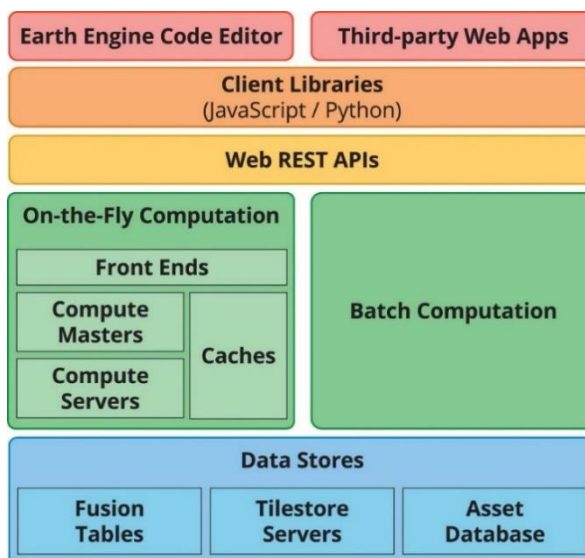
Dataset	Nominal resolution	Temporal granularity	Temporal coverage	Spatial coverage
Landsat				
Landsat 8 OLI/TIRS	30 m	16 day	2013–Now	Global
Landsat 7 ETM +	30 m	16 day	2000–Now	Global
Landsat 5 TM	30 m	16 day	1984–2012	Global
Landsat 4–8 surface reflectance	30 m	16 day	1984–Now	Global
Sentinel				
Sentinel 1 A/B ground range detected	10 m	6 day	2014–Now	Global
Sentinel 2A MSI	10/20 m	10 day	2015–Now	Global
MODIS				
MOD08 atmosphere	1°	Daily	2000–Now	Global
MOD09 surface reflectance	500 m	1 day/8 day	2000–Now	Global
MOD10 snow cover	500 m	1 day	2000–Now	Global

Hình 1.10 - Một số bộ dữ liệu sẵn có trên Google Earth Engine.

- Sức mạnh tính toán (Compute power): Cơ sở hạ tầng tính toán của Google, được tối ưu hóa để xử lý song song dữ liệu không gian địa lý.
- APIs: Các API dành cho JavaScript and Python (được lưu trữ trên GitHub) có thể dùng để gửi yêu cầu đến máy chủ Earth Engine.
- Code Editor: Môi trường phát triển tích hợp (IDE) trực tuyến để tạo mẫu nhanh và trực quan hóa các phân tích không gian phức tạp bằng API Javascript.

❖ Kiến trúc hệ thống GEE.

Earth Engine Code Editor và các ứng dụng của bên thứ ba sử dụng thư viện dành cho máy khách (client libraries) để gửi các truy vấn tương tác đến hệ thống thông qua REST API. Các yêu cầu gửi đến được xử lý ngay bởi các máy chủ Front End hoặc chuyển tiếp các yêu cầu phụ phức tạp đến các máy chủ Compute Master và Compute Server để tiếp tục tính toán lưu trữ phân tán trên hạ tầng điện toán đám mây của Google.



Hình 1.11 - Mô hình đơn giản hóa hệ thống Google Earth Engine.

1.5.4.2 Google Cloud Platform.

Google Cloud Platform (GCP), được cung cấp bởi Google, là một bộ dịch vụ điện toán đám mây chạy trên cùng một cơ sở hạ tầng mà Google. Bên cạnh một bộ công cụ quản lý, nó cung cấp một loạt các dịch vụ đám mây bao gồm điện toán, lưu trữ dữ liệu, phân tích dữ liệu và học máy.

1.6 Tổng kết chương 1

Trong chương này, tác giả đã tóm tắt sơ lược một cách ngắn gọn những nghiên cứu về RVB, hiện trạng RVB Việt Nam và nguyên nhân biến động RVB; khái quát về tính khả thi của việc giám sát biến động bằng ảnh vệ tinh, chỉ ra khó khăn thách thức. Đồng thời, cũng có cái nhìn tổng quan về việc ứng dụng trí tuệ nhân tạo, học máy, mạng nơ-ron vào việc giải quyết bài toán phát hiện mất rừng từ ảnh vệ tinh. Cuối cùng khái quát ngắn gọn một số kiến thức và công nghệ liên quan sử dụng trong nghiên cứu.

CHƯƠNG 2

PHÁT HIỆN MẮT RỪNG VÉN BIÊN

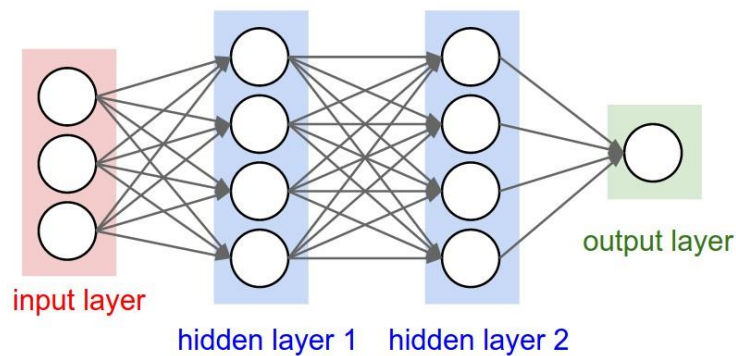
TỪ ẢNH VỆ TINH DỰA TRÊN MẠNG NƠ-RON

2.1 Mạng nơ-ron.

Mạng nơ-ron nhân tạo (ANN), gọi tắt là mạng nơ-ron (Neural Network - NN) là một mô hình toán học được xây dựng dựa trên cơ sở các mạng nơ-ron sinh học gồm một số lượng lớn các phần tử (gọi là nơ-ron) kết nối với nhau thông qua các liên kết (gọi là trọng số liên kết) làm việc như một thể thống nhất để giải quyết các vấn đề.

Kết hợp với các kĩ thuật DL, NN đang trở thành một công cụ rất mạnh mẽ mang lại hiệu quả tốt nhất cho nhiều bài toán khó như nhận dạng ảnh, giọng nói hay xử lý ngôn ngữ tự nhiên...vv

Một mạng NN tổng quát là mạng có n lớp ($n > 2$) trong đó bao gồm một lớp đầu vào, một đầu lớp ra và một hoặc nhiều lớp ẩn.

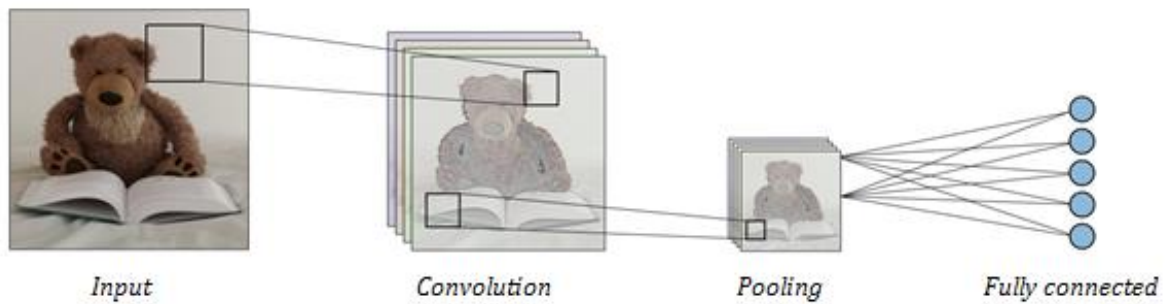


2.2 Mạng nơ-ron tích chập.

Kiến trúc mạng CNN là kiến trúc mở rộng của mạng NN được sử dụng rộng rãi trong kỹ thuật DL đặc biệt là trong lĩnh vực thị giác máy tính (computer vision). Một trong những giải pháp nhằm tối ưu quá trình huấn luyện mạng được đề xuất là giảm số lượng các trọng số (weight) để tăng tốc độ tính toán, giảm thời gian huấn luyện, tránh hiện tượng quá khớp (overfitting) khi mà lượng dữ liệu đầu vào là rất lớn như các bức ảnh màu, video..vv

2.2.1 Giới thiệu tổng quan.

Mạng nơ-ron tích chập (CNN) là một trong những thuật toán DL cho kết quả tốt nhất hiện nay trong hầu hết các bài toán về thị giác máy như phân lớp, nhận dạng... Một dạng mạng CNN được cấu thành bởi các tầng sau: Convolution, Pooling và Fully Connected.



Hình 2.1 - Các thành phần cấu thành mạng nơ-ron tích chập (CNN).

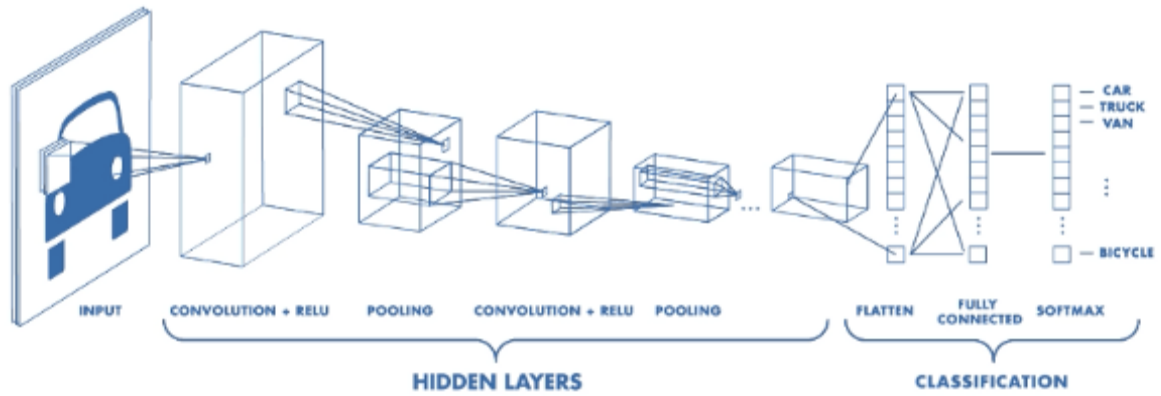
Như vậy ta có thể thấy một CNN về cơ bản có 3 quá trình khác nhau:

- **Convolution:** Thông qua các tích chập giữa ma trận đầu vào với bộ lọc để tạo thành các đơn vị trong một lớp mới. Quá trình có thể diễn ra liên tục ở phần đầu của mạng và thường sử dụng hàm kích hoạt ReLU. Quá trình này được gọi là trích chọn đặc trưng (convolution).

- **Pooling:** Các lớp ở sau quá trình trích chọn đặc trưng sẽ có kích thước lớn do số đơn vị ở các lớp sau thường tăng tiến theo cấp số nhân. Điều đó làm tăng số lượng hệ số và khối lượng tính toán trong mạng nơ-ron. Vì vậy để giảm tải tính toán chúng ta sẽ cần giảm chiều của ma trận hoặc giảm số đơn vị của lớp.

Tuy nhiên mỗi một đơn vị sẽ là kết quả đại diện của việc áp dụng một bộ lọc để tìm ra một đặc trưng cụ thể nên việc giảm số đơn vị sẽ không khả thi. Giảm kích thước ma trận thông qua việc tìm ra một giá trị đại diện cho mỗi một vùng không gian mà bộ lọc đi qua sẽ không làm thay đổi các đường nét chính của bức ảnh nhưng lại giảm được kích thước của ảnh. Do đó quá trình giảm chiều ma trận được áp dụng. Quá trình này gọi là tổng hợp hoặc gộp (pooling).

- **Fully Connected:** Sau khi đã giảm số lượng tham số đến một mức độ hợp lý, ma trận cần được làm phẳng (flatten) thành một vector và sử dụng các kết nối hoàn toàn giữa các lớp. Quá trình này sẽ diễn ra ở cuối mạng tích chập và sử dụng hàm kích hoạt là ReLU. Kết nối cuối cùng sẽ dẫn tới các đơn vị là đại diện cho mỗi lớp với hàm kích hoạt là Softmax nhằm mục đích tính xác suất.



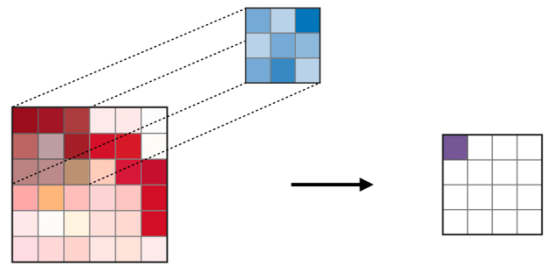
Hình 2.2 - Cấu trúc đại diện của một mạng nơ-ron tích chập đầy đủ.

Tầng tích chập (convolution) và tầng hợp nhất (pooling) có thể được hiệu chỉnh theo các siêu tham số (hyperparameters) được mô tả ở những phần tiếp theo.

2.2.2 Các kiểu tầng.

❖ Tầng tích chập (CONV).

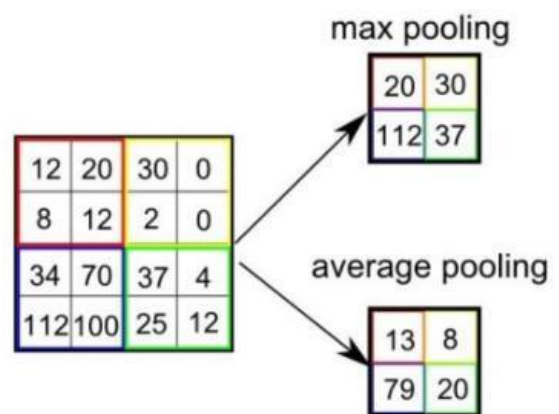
Tầng tích chập (CONV) sử dụng các bộ lọc để thực hiện phép tích chập khi đưa chúng đi qua đầu vào I theo các chiều của nó. Các siêu tham số của các bộ lọc này bao gồm kích thước bộ lọc F và độ trượt (stride) S . Kết quả đầu ra O được gọi là feature map hay activation map.



Hình 2.3 - Ví dụ phép tích chập.

❖ Tầng hợp nhất (POOL).

Tầng hợp nhất (POOL) là một phép downsampling, thường được sử dụng sau tầng tích chập, dùng để giảm chiều của ma trận đầu vào nhưng vẫn làm nổi bật lên được đặc trưng. Cụ thể, max pooling và average pooling là những dạng pooling thường được sử dụng, mà tương ứng là trong đó giá trị lớn nhất và giá trị trung bình được lấy ra.



Hình 2.4 - Ví dụ max pooling và average pooling.

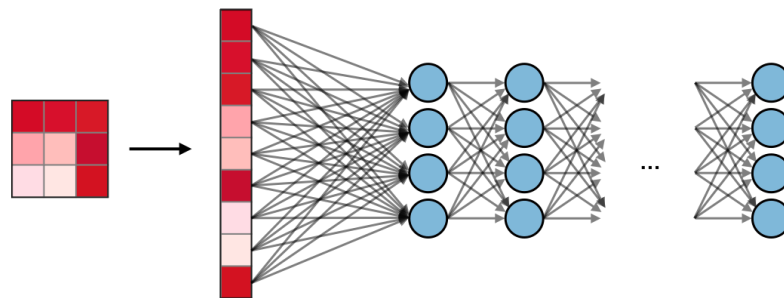
So sánh max pooling và average pooling:

Kiểu	Max pooling	Average pooling
Chức năng	Từng phép pooling chọn giá trị lớn nhất trong khu vực mà nó đang được áp dụng	Từng phép pooling tính trung bình các giá trị trong khu vực mà nó đang được áp dụng
Minh họa		
Nhận xét	<ul style="list-style-type: none"> - Bảo toàn các đặc trưng đã phát hiện - Được sử dụng thường xuyên 	<ul style="list-style-type: none"> - Giảm kích thước feature map - Được sử dụng trong mạng LeNet

Bảng 2.1 - So sánh max pooling và average pooling.

❖ Tầng kết nối đầy đủ (FC)

Tầng kết nối đầy đủ (FC) nhận đầu vào là các dữ liệu đã được làm phẳng, mà mỗi đầu vào đó được kết nối đến tất cả nơ-ron. Trong mô hình mạng CNN, các tầng kết nối đầy đủ thường được tìm thấy ở cuối mạng và được dùng để tối ưu hóa mục tiêu của mạng ví dụ như độ chính xác của lớp.

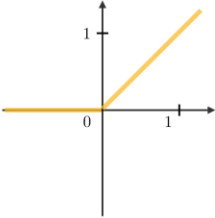
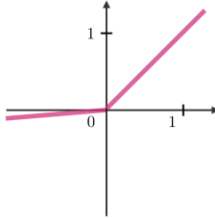
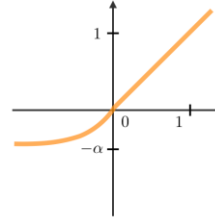


Hình 2.5 – Mô tả tầng kết nối đầy đủ (Fully connected).

2.2.3 Các hàm kích hoạt thường sử dụng.

❖ Rectified Linear Unit

Tầng rectified linear unit (ReLU) là một hàm kích hoạt g được sử dụng trên tất cả các thành phần. Mục đích của nó là tăng tính phi tuyến tính cho mạng. Những biến thể khác của ReLU được tổng hợp ở bảng dưới:

ReLU	Leaky ReLU	ELU
$g(z)=\max(0,z)$	$g(z)=\max(\epsilon z, z)$ với $\epsilon \ll 1$	$g(z)=\max(\alpha(e^z-1), z)$ với $\alpha \ll 1$
		
Độ phức tạp phi tuyến tính có thể thông dịch được về mặt sinh học	Gán vấn đề ReLU chết cho những giá trị âm	Khả vi tại mọi nơi

Bảng 2.2 - So sánh cách biến thể của hàm kích hoạt ReLU

❖ Softmax

Bước softmax có thể được coi là một hàm logistic tổng quát lấy đầu vào là một vector chứa các giá trị $x \in \mathbb{R}^n$ và cho ra là một vector gồm các xác suất $p \in \mathbb{R}^n$ thông qua một hàm softmax ở cuối kiến trúc. Nó được định nghĩa như sau:

$$p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} \quad \text{với} \quad p_i = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

Hình 2.6 - Hàm softmax

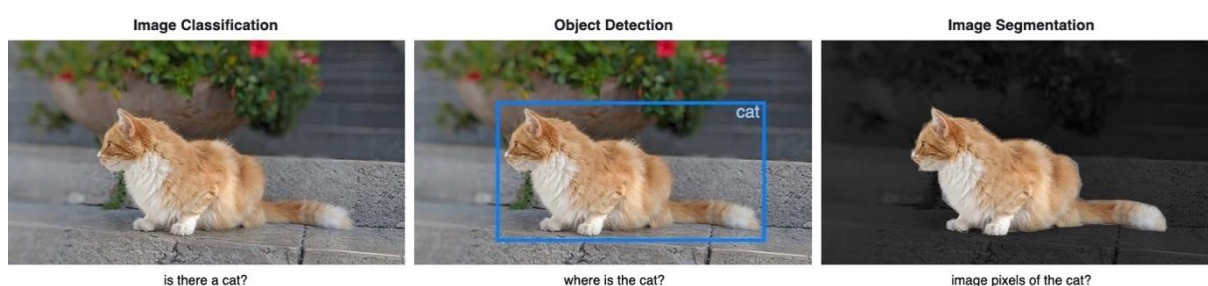
2.3 Kiến trúc mạng U-Net.

2.3.1 Hiểu về phân loại hình ảnh, phát hiện đối tượng và phân đoạn hình ảnh.

Phân loại hình ảnh giúp chúng ta phân loại những gì có trong một hình ảnh. Mục đích là để trả lời, có con mèo trong hình ảnh này không, bằng cách dự đoán có hoặc không.

Phát hiện đối tượng chỉ định vị trí của các đối tượng trong ảnh. Mục tiêu là xác định được con mèo trong hình ảnh này ở đâu? Cảnh, bằng cách vẽ một hộp giới hạn xung quanh đối tượng quan tâm.

Phân đoạn hình ảnh tạo ra mặt nạ điểm ảnh (pixel) của từng đối tượng trong ảnh. Mục tiêu là xác định vị trí và hình dạng của các đối tượng khác nhau trong hình ảnh bằng cách phân loại từng pixel trong nhãn mong muốn.

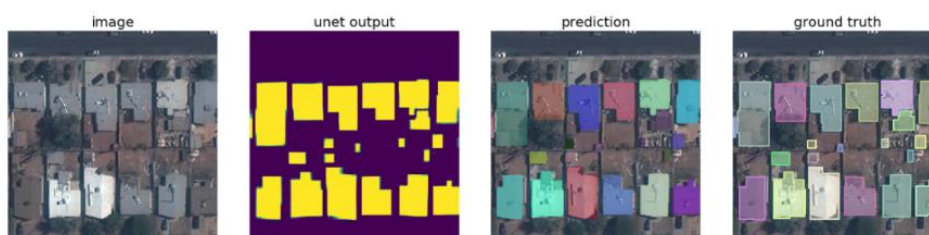


Hình 2.7 - Phân loại hình ảnh, phát hiện đối tượng và phân đoạn hình ảnh.

2.3.2 Giới thiệu U-Net.

U-Net¹² là một kiến trúc mạng nơ-ron tích chập giúp phân đoạn nhanh và chính xác hình ảnh. Cho đến nay, U-Net đã vượt trội so với phương pháp tốt nhất trước đây, thậm chí nó còn hiệu quả với hình ảnh dữ liệu hạn chế. Nó đã chiến thắng Grand Challenge về Máy tính tự động phát hiện sâu răng trong ảnh X-Quang tại ISBI 2015 và chiến thắng Thử thách theo dõi tế bào tại ISBI 2015.

U-Net lần đầu tiên được đề xuất bởi Ronneberger, Phillip Fischer và Thomas Brox vào năm 2015. Nó được thiết kế và sử dụng nhiều trong các nhiệm vụ phân đoạn hình ảnh y sinh nhưng nó cũng hoạt động để phân đoạn hình ảnh tự nhiên, ảnh viễn thám.

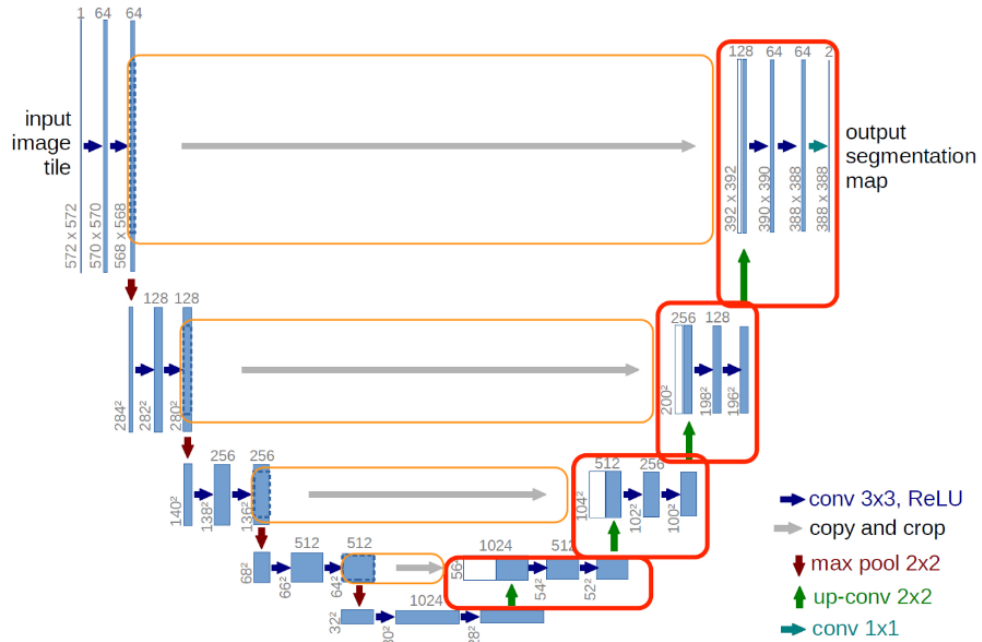


Hình 2.8 - Ứng dụng U-Net trong phân đoạn và dự đoán ảnh viễn thám.

¹² <https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/>

2.3.3 Kiến trúc mạng U-Net.

U-Net được đặt tên theo đặc điểm kiến trúc khi được trực quan hóa.



Hình 2.9 - Kiến trúc mạng U-Net.

❖ Mạng U-Net gồm:

- Đường dẫn trái (còn gọi là đường dẫn co - contraction path): Giống như một kiến trúc CNN điển hình, bằng cách xếp liên tiếp hai tích chập Conv 3x3 (mũi tên xanh dương) theo sau là một tổng hợp max pooling 2x2 (mũi tên đỏ) để lấy mẫu xuống. Ở mỗi bước lấy dữ liệu xuống (downsampling), số lượng kênh được nhân đôi.

- Đường dẫn phải (còn gọi là đường dẫn mở rộng - expansion path): Một tích chập Conv 2x2 (mũi tên xanh lục) lấy mẫu lên và hai tích chập Conv 3x3 (mũi tên xanh dương). Ở mỗi bước đưa dữ liệu lên (upsampling), số lượng kênh được giảm một nửa.

- Tầng cuối cùng: Một tích chập Conv 1x1 để ánh xạ bản đồ đặc trưng đến số lớp mong muốn.

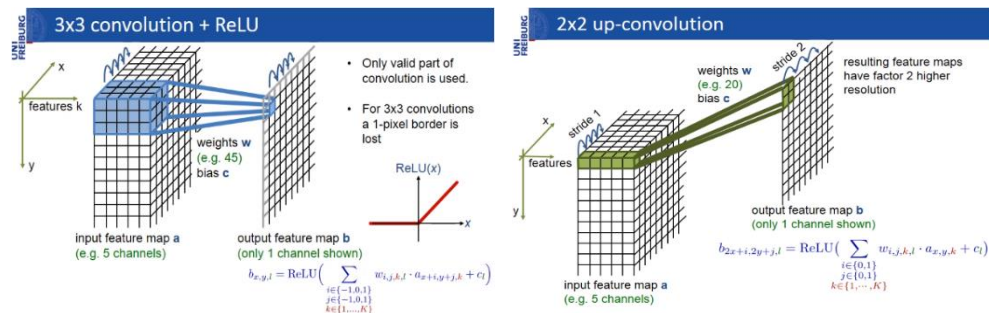
❖ Xác định mất mát.

Mất mát có thể được tính bằng binary cross-entropy và dice loss. Đây là tiêu chí đánh giá hiệu suất được sử dụng thường xuyên để đánh giá và áp dụng thành công trong phân đoạn hình ảnh y sinh.

$$L_{bce} = \sum_i y_i \log o_i + (1 - y_i) \log (1 - o_i) \quad L_{Dice} = -\frac{2 \sum_i o_i y_i}{\sum_i o_i + \sum_i y_i}$$

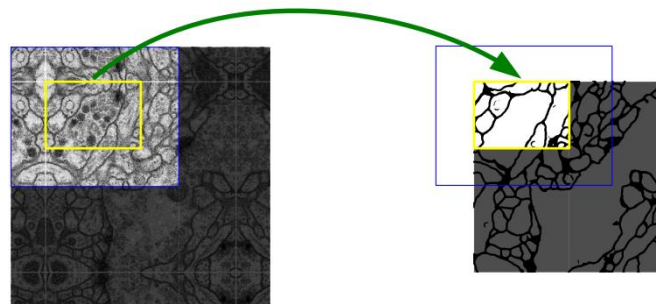
2.3.4 Đặc điểm mạng U-Net.

Toàn bộ kiến trúc sử dụng các tầng tích chập (CONV), mỗi tích chập tiêu chuẩn được kích hoạt bởi một hàm kích hoạt ReLU và không hề sử dụng tầng kết nối đầy đủ (FC) nào cả. Thay vào đó việc kết nối các đặc trưng sẽ do nửa thứ hai của "chữ U" đảm nhận, điều này giúp mạng không cần tầng FC. Do đó, U-Net có thể chấp nhận đầu vào với kích thước bất kì.



Hình 2.10 - Mô tả cho một quá trình tích chập và giải mã trong U-Net

U-Net sử dụng phương pháp đệm (padding method) thêm điểm ảnh (add pixel), điều này giúp kiến trúc có thể phân đoạn hình ảnh được hoàn toàn. Phương pháp này đặc biệt quan trọng khi phân đoạn cho các hình ảnh; nếu không, độ phân giải có thể bị hạn chế bởi dung lượng của bộ nhớ GPU.



Hình 2.11 - Phương pháp padding method.

U-Net có hai ưu điểm chính:

- Yêu cầu ít mẫu huấn luyện.
- Phân đoạn nhanh và chính xác.

Dựa trên, những kết quả đã công bố, cùng các đặc điểm và ưu điểm của kiến trúc mạng U-Net cho thấy đây là kiến trúc phù hợp giúp giải bài toán “Phát hiện biến động rừng ven biển từ ảnh vệ tinh”.

2.4 Phát hiện mất rừng từ ảnh vệ tinh dựa trên mạng U-Net

2.4.1 Mô tả bài toán.

Mỗi tư liệu ảnh chụp từ vệ tinh tại mỗi thời điểm là khác nhau, mỗi ảnh sẽ chứa đựng trong nó những đặc trưng khác nhau. Đặc trưng đó trong ảnh quang học chính là đặc trưng quang phổ. Việc kết hợp nhiều ảnh chụp khác nhau sẽ thu được cảnh lớn bao phủ toàn bộ phạm vi mong muốn giám sát, nếu chồng xếp hai hoặc nhiều cảnh ảnh khác nhau tại cùng một vị trí ta sẽ thu được bộ dữ liệu ảnh giám sát.

Chúng ta có thể xác định được những vị trí bất thường, phát hiện biến động rừng bằng việc phân tích bộ dữ liệu ảnh giám sát ở trên. Tuy nhiên, quá trình này thường mất nhiều thời gian, cần người thực hiện phải có kiến thức chuyên môn, đối chiếu liên tục với các dữ liệu liên quan khác để đi đến kết luận cuối cùng về vị trí bất thường đó là mất rừng hay tăng rừng.

Tư liệu ảnh tăng, phạm vi giám sát tăng lên, đối tượng quan tâm nhiều lên điều đó có nghĩa chúng ta đang tạo ra khối dữ liệu vô cùng lớn, việc sử dụng con người làm trung tâm trong chuỗi quy trình giám sát biến động sẽ không còn khả thi nữa.

Ứng dụng trí tuệ nhân tạo (AI), sử dụng mạng thần kinh nhân tạo chính là giải pháp hướng để giải quyết bài toán. Làm cho máy tính có thể học hỏi, tư duy như con người để giải quyết bài toán như các con người xử lý.

Và bài toán phát hiện biến động rừng từ ảnh vệ tinh đưa về bài toán phân loại hình ảnh, phân mảnh xác định đối tượng. Bài toán được hiểu như sau:

- **Đầu vào:**

- Dữ liệu ảnh vệ tinh: Được khai thác và cập nhật mới nhất từ GEE; Dữ liệu ảnh tiếp tục trải qua các bước tiền xử lý ảnh (hiệu chỉnh, loại bỏ mây, tính toán các chỉ số..vv)
- Dữ liệu hiện trạng rừng: Được khai thác từ Hệ thống thông tin quản lý ngành Lâm nghiệp - FORMIS;
- Dữ liệu điểm mẫu: Bộ dữ liệu tọa độ vị trí các điểm huấn luyện (training) và xác minh (validation);

- **Đầu ra:**

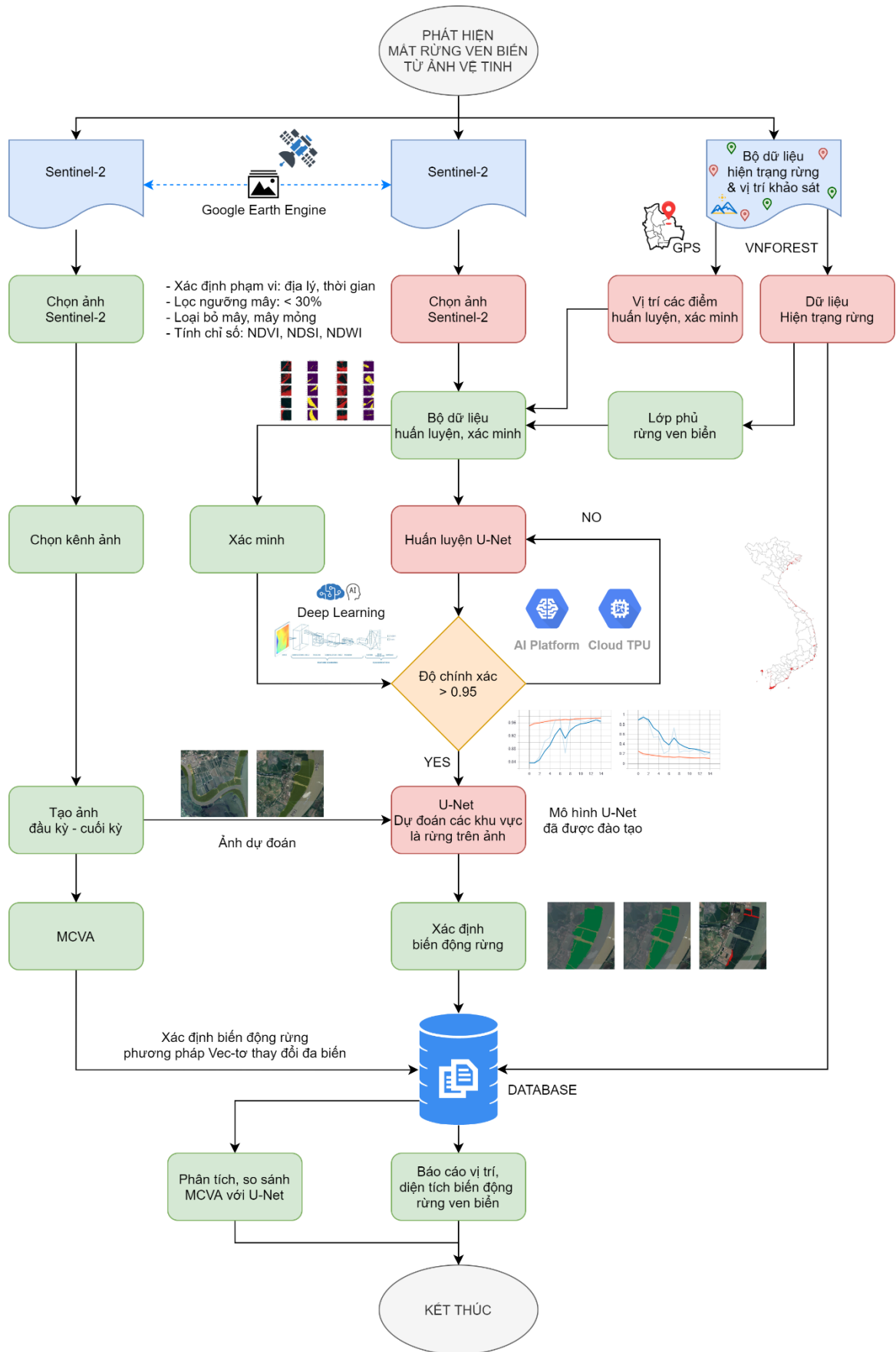
- Mô hình dự đoán đã được tối ưu.
- Lớp phủ rừng ven biển từ ảnh dự đoán.
- Vị trí biến động tăng rừng hoặc giảm rừng.

2.4.2 Quy trình thực hiện.

Quy trình thực hiện có 5 bước quan trọng, gồm:

- [1] Lấy tư liệu ảnh *Sentinel-2*, dữ liệu hiện trạng rừng và dữ liệu điểm mẫu.
 - Tư liệu ảnh *Sentinel-2* được khai thác từ *GEE* và tiền xử lý;
 - Dữ liệu hiện trạng rừng được kế thừa từ hệ thống thông tin quản lý ngành Lâm nghiệp và xử lý chuyển đổi thành ảnh lớp phủ rừng;
 - Dữ liệu điểm mẫu được kế thừa từ hoạt động khảo sát thực địa.
- [2] Xây dựng bộ dữ liệu huấn luyện.
 - Dữ liệu ảnh vệ tinh và lớp phủ đã được xử lý tại bước 1 sẽ được lấy theo từng điểm mẫu (xem phương pháp xử lý tại mục 3.3.2.3).
 - Bộ dữ liệu huấn luyện sẽ chứa đầy đủ thông tin đặc trưng của dữ liệu đầu vào.
 - Bộ dữ liệu huấn luyện được tạo từ các điểm mẫu huấn luyện; bộ dữ liệu xác minh được tạo từ các điểm mẫu xác minh.
- [3] Huấn luyện mạng U-Net.
- [4] Sử dụng mô hình đã huấn luyện để dự đoán kiểm tra.
- [5] Xác định biến động rừng, biến động mất rừng.

Sơ đồ quy trình thực hiện:



Hình 2.12 - Quy trình phát hiện biến động rừng từ ảnh vệ tinh dựa trên mạng nơ-ron.

2.5 Tổng kết chương 2

Trong chương này, luận văn đã trình bày tóm tắt về mạng nơ-ron tích chập và kiến trúc mạng U-Net. Trên cơ sở những đặc điểm, ưu điểm và kết quả nghiên cứu đã công bố để chọn mô hình mạng nơ-ron phù hợp cho bài toán phát hiện biến động rừng. Đồng thời mô tả cụ thể bài toán, quy trình các bước thực hiện giải bài toán “Phát hiện biến động mất rừng từ ảnh vệ tinh dựa trên mạng nơ-ron”.

CHƯƠNG 3

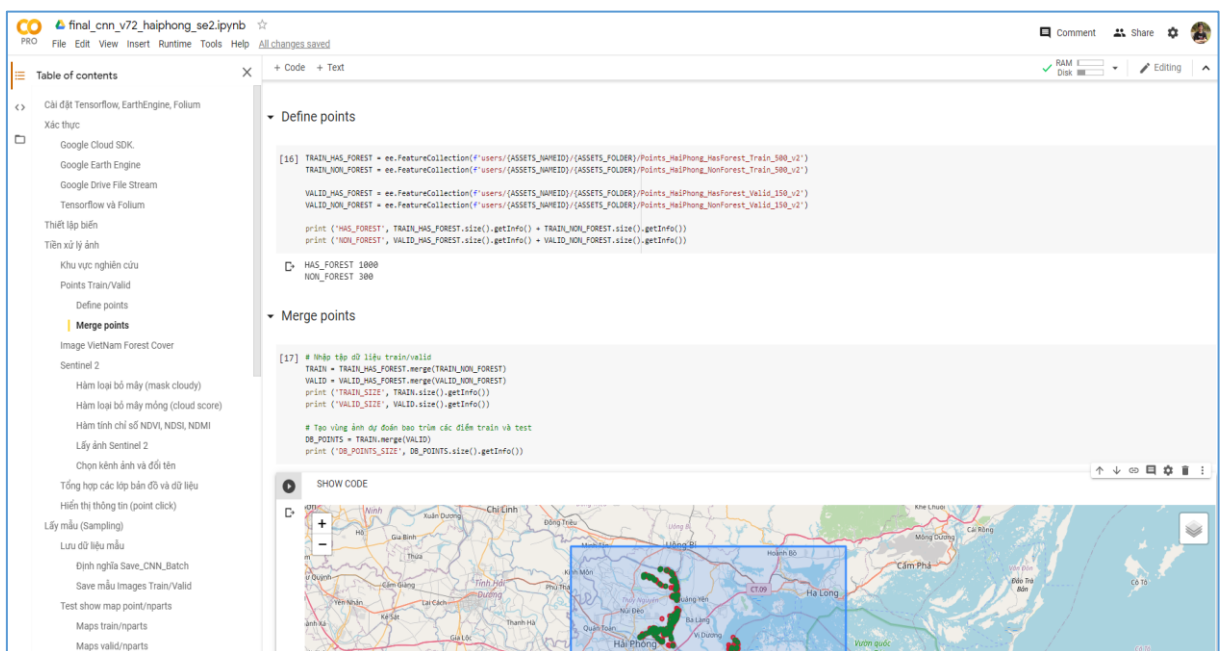
THỬ NGHIỆM VÀ ĐÁNH GIÁ

3.1 Phần mềm và thư viện.

Để triển khai kiến trúc mạng U-Net và tương tác với dịch vụ GEE, cũng như khai thác tối đa sức mạnh tính toán nền tảng điện toán đám mây của Google tôi tiến hành tạo một dự án thử nghiệm trên Google Colaboratory và cài đặt trên đó một số thư viện cần thiết.

3.1.1 Phần mềm.

- Python: phiên bản 3.6.8
- Google Colaboratory: gọi tắt là Colab là một sản phẩm từ Google Research. Colab cho phép bất cứ ai viết và thực thi mã python tùy ý thông qua trình duyệt và đặc biệt phù hợp với học máy và phân tích dữ liệu. Thông tin chi tiết Colab sử dụng:
 - Phiên bản: Colab Pro.
 - Hình dạng thời gian chạy: High-RAM.
 - Bộ xử lý tăng tốc tính toán: TPU¹³ (Tensor Processing Unit).



Hình 3.1 - Giao diện phần mềm Colab.

¹³ TPU là một mạch tích hợp dành riêng cho ứng dụng tăng tốc AI.

3.1.2 Thư viện.

- Tensorflow: là một nền tảng học máy nguồn mở đa dụng và có độ linh hoạt cao phục vụ việc xây dựng các ứng dụng trí tuệ nhân tạo;
- EarthEngine-API: là thư viện nền tảng xử lý không gian địa lý dựa trên đám mây tiên tiến nhất trên thế giới.
- Folium: là thư viện giúp hiển thị và tương tác với bản đồ qua python.

3.2 Chuẩn bị dữ liệu.

3.2.1 Vị trí thử nghiệm.

Địa điểm thử nghiệm: phạm vi rừng ven biển tại thành phố Hải Phòng.

Ranh giới hành chính:

- Phía Bắc giáp tỉnh Quảng Ninh; Phía Nam giáp tỉnh Thái Bình
- Phía Tây giáp tỉnh Hải Dương; Phía Đông giáp biển Đông.

Thành phố có tọa độ địa lý:

- Từ 20030'39' - 21001'15' Vĩ độ Bắc.
- Từ 106023'39' - 107008'39' Kinh độ Đông.
- Và huyện đảo Bạch Long Vỹ nằm giữa Vịnh Bắc Bộ, có tọa độ :
 - Từ 20007'35' - 20008'36' Vĩ độ Bắc và
 - Từ 107042'20' - 107044'15' Kinh độ Đông.



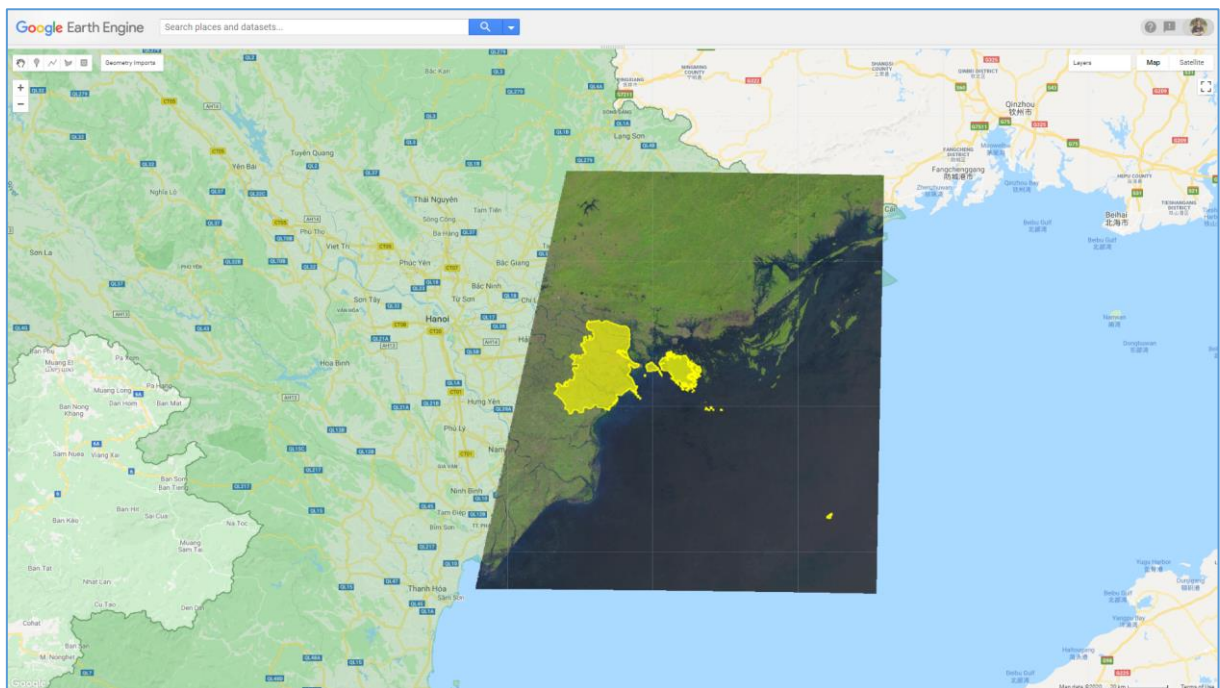
Hình 3.2 - Bản đồ thành phố Hải Phòng

3.2.2 Dữ liệu ảnh Sentinel-2

Đề tài khai thác sử dụng Sentinel-2 từ bộ dữ liệu “*Sentinel-2 MSI: MultiSpectral Instrument, Level-2A*” sẵn có từ 28/03/2017 đến nay trên Google Earth Engine. Dữ liệu ảnh chọn tham gia huấn luyện và xác minh được chụp năm 2018; dữ liệu ảnh sử dụng để dự đoán được chụp năm 2019.

Bộ dữ liệu ảnh tiếp tục trải qua các bước tiền xử lý trước khi sử dụng, gồm:

- Chọn cảnh ảnh bao phủ phạm vi giám sát.
- Chọn các kênh ảnh: B12, B11, B8A, B8, B7, B6, B5, B4, B3, B2.
- Lọc và chọn các cảnh ảnh chụp thỏa mãn phạm vi thời gian.
- Lọc và chọn các cảnh ảnh có ngưỡng mây nhỏ hơn 30%.
- Loại bỏ mây, bóng mây và dải mây mỏng.
- Tính toán và thêm các chỉ số khác biệt chuẩn: NDVI, NDSI, NDVI.
- Lấy trung vị (median) bộ sự tập ảnh.
- Cắt ảnh theo phạm vi giám sát



Hình 3.3 - Bản đồ cảnh ảnh sử dụng đã loại bỏ mây và bóng mây.

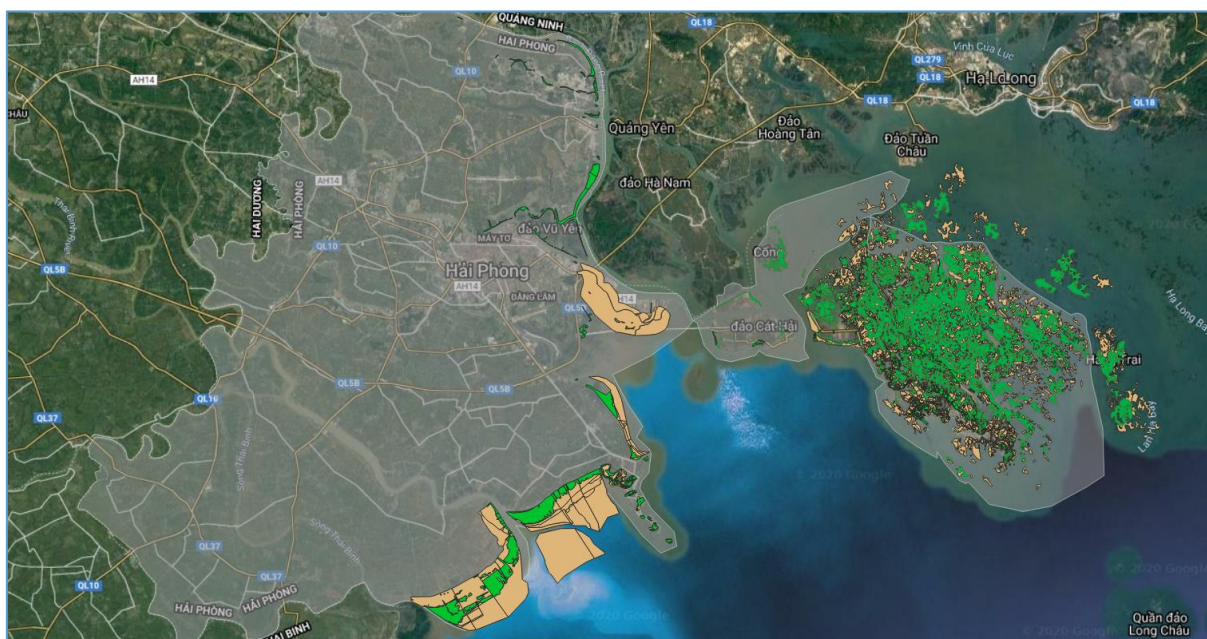
Cụ thể, số lượng ảnh đã được chọn tham gia huấn luyện và xác minh được chụp năm 2018 là 26 cảnh ảnh, số lượng ảnh sử dụng dự đoán năm 2019 là 32 cảnh ảnh.

Chi tiết xem thêm tại phụ lục 1.

3.2.3 Dữ liệu bản đồ hiện trạng rừng.

Đề tài kế thừa dữ liệu từ “Hệ thống thông tin quản lý ngành Lâm nghiệp - FORMIS”, gồm các lớp dữ liệu sau:

- Lớp dữ liệu bản đồ ranh giới cấp hành chính¹⁴ Việt Nam.
- Lớp dữ liệu bản đồ hiện trạng rừng ven biển Hải Phòng¹⁵.



Hình 3.4 - Bản đồ thể hiện hiện trạng rừng trên nền ranh giới thành phố Hải Phòng.

Hiện trạng diện tích các loại rừng ven biển và đất lâm nghiệp phân theo mục đích sử dụng của thành phố Hải Phòng theo công bố của Tổng cục Lâm nghiệp Việt Nam là 26.127,58¹⁶ hecta (Quyết định 423/QĐ-TCLN-PTR ngày 31/12/2019).

TT	Phân loại rừng	Tổng diện tích	Đặc dụng	Phòng hộ	Sản xuất
	TỔNG	26,127.58	9,931.60	16,123.24	72.74
A	DIỆN TÍCH CÓ RỪNG	11,331.29	5,984.28	5,274.27	72.74
I	Rừng tự nhiên	9,000.72	5,779.54	3,182.75	38.43
II	Rừng trồng	2,330.57	204.74	2,091.52	34.31
B	DIỆN TÍCH CHƯA THÀNH RỪNG	14,796.29	3,947.32	10,848.97	-
I	Diện tích trồng chưa thành rừng	785.31	-	785.31	-
II	Diện tích khoanh nuôi tái sinh	4,238.89	1,948.01	2,290.88	-
III	Diện tích khác	9,772.09	1,999.31	7,772.78	-

Bảng 3.1 - Thông kê diện tích hiện trạng RVB TP. Hải Phòng (31/12/2019).

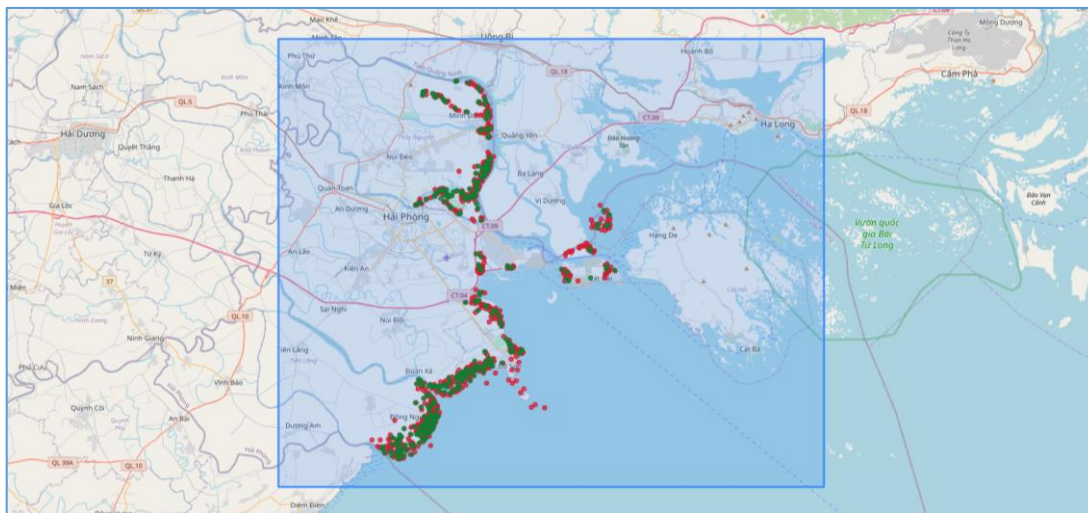
¹⁴ Chi tiết dữ liệu ranh giới hành chính Việt Nam: https://bit.ly/ranhgioi_hanhchinh_vietnam

¹⁵ Chi tiết hiện trạng rừng ven biển TP. Hải Phòng: https://bit.ly/hientrangrung_haiphong

¹⁶ Nguồn dữ liệu: <http://rungvenbien.ifee.edu.vn/thongkebaocao>

3.2.4 Dữ liệu điểm mẫu.

Dữ liệu điểm mẫu là bộ dữ liệu gồm 1000 điểm huấn luyện (training) và 300 điểm xác minh (validation). Bộ dữ liệu mẫu này được kế thừa từ kết quả nhiệm vụ xây dựng Bộ cơ sở dữ liệu RVB Việt Nam của đội ngũ chuyên gia Viện Sinh thái rừng và Môi trường, Trường Đại học Lâm nghiệp. Chi tiết dữ liệu điểm mẫu được công khai tại đường dẫn: https://bit.ly/hoatx_points_dataset_train_valid



Hình 3.5 - Bản đồ hiển thị các điểm mẫu khóa ảnh huấn luyện và xác minh.

Mỗi vị trí điểm mẫu xác minh cho biết các thông tin: thông tin hành chính cấp xã/phường, tọa độ vị trí, trạng thái rừng (có rừng bằng 1 hoặc không có rừng bằng 0) và thông tin tiểu khu – khoảnh – lô.

STT	Xã/Phường	Tọa độ X	Tọa độ Y	Trạng thái	Tiểu khu/Khoảnh/Lô
1	Đông Hưng	106.64850	20.62231	1	T.L/9/5
2	Đại Hợp	106.73653	20.69108	0	K.T/4/2
3	Phục Lễ	106.76173	20.91027	1	11542/23/3a
4	Ngọc Hải	106.79105	20.72424	1	DS/2a/3
5	Ngọc Hải	106.79147	20.72120	0	DS/2a/4
6	Văn Phong	106.87590	20.79599	0	2A/25/1
...

Bảng 3.2 - Mẫu thu thập dữ liệu điểm mẫu khóa xác nhận tại thực địa.

3.3 Cài đặt thực nghiệm.

3.3.1 Cài đặt thư viện và thiết lập chung.

❖ Cài đặt thư viện:

Để bắt đầu thử nghiệm, đầu tiên tiến hành cài đặt các thư viện cần thiết.

```
!pip install tf-nightly # Tensorflow
!pip install tf-nightly-gpu
!pip install earthengine-api # EarthEngine-API
!pip install folium # Folium
%load_ext tensorboard # Tensorboard
```

Thông tin phiên bản các thư viện

```
Tensorflow 2.2.0
Tensorboard 2.2.2
Keras 2.3.1
Folium 0.8.3
```

❖ Thiết lập chung.

Tiếp theo, thực hiện cài đặt một số biến chung cho toàn bộ quá trình thử nghiệm: thời gian bắt đầu - kết thúc, đây là thời gian xác định dùng để chọn ra các cảnh ảnh phù hợp nằm trong phạm vi thời gian nghiên cứu và phù hợp với dữ liệu hiện trạng rừng khai thác từ “Hệ thống thông tin quản lý ngành Lâm nghiệp - FORMIS”. Và một số biến dùng chung như: ngưỡng mây, thông tin các kênh ảnh sử dụng, độ phân giải..vv

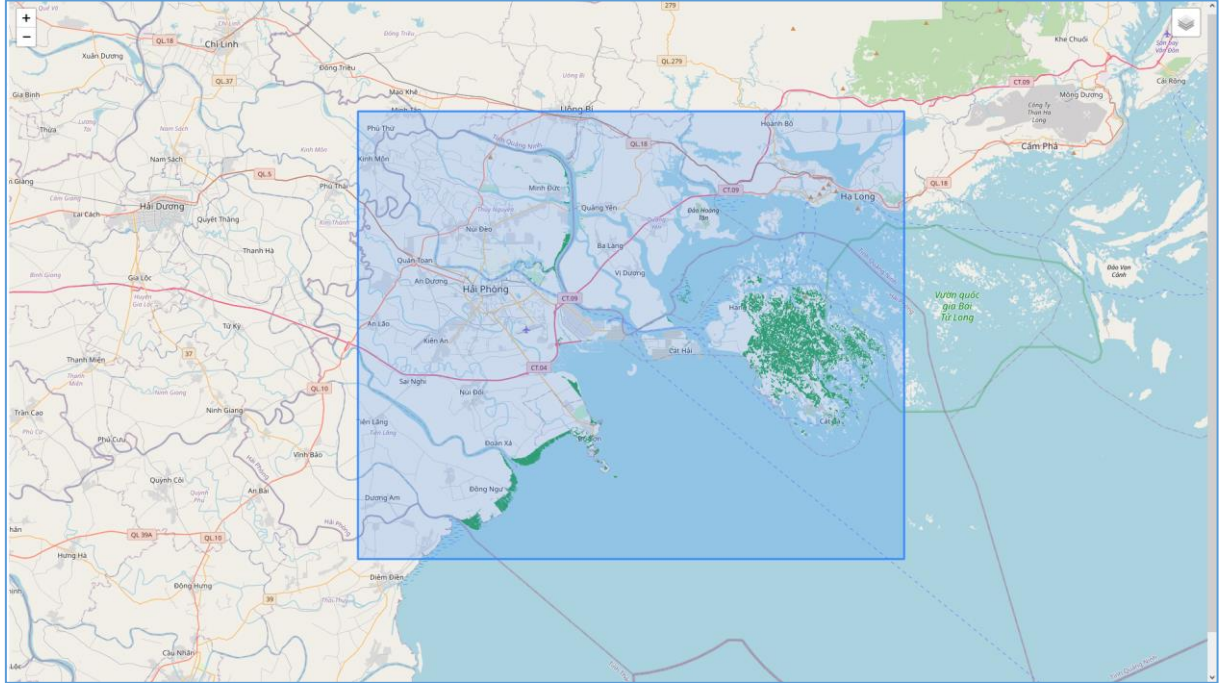
```
MAP_ZOOM = 11
START_DAY = '2018-01-01' # Ngày bắt đầu
END_DAY = '2018-12-31' # Ngày kết thúc
CLOUD_PERCENT = 30 # Tỷ lệ phần trăm mây
SCALE = 10 # Độ phân giải ảnh
KERNEL_SIZE = 128 # Kích thước kernel
FOLDER = f'unet_forest' # Thư mục lưu trữ
```

Tiếp theo, tôi xác định phạm vi khu vực thử nghiệm.

```
REGION_MAP = ee.FeatureCollection(
    [ee.Feature(
        ee.Geometry.Polygon(
            [[[106.53601110780845, 21.038767235681952],
              [106.53601110780845, 20.583057129310447],
              [107.13064611757407, 20.583057129310447],
              [107.13064611757407, 21.038767235681952]]]],
            {
                "forest": 0,
                "system:index": "0"
            })])])
```

Tiếp theo, chỉ định lớp phủ rừng ven biển và chỉ định nguồn ảnh Sentinel-2.

```
# Lớp phủ rừng ven biển
VNFC = ee.Image('users/trhoa06/vietnam_forest_cover/VNFC_HaiPhong')
# Bộ sưu tập ảnh Sentinel-2 từ Google Earth Engine
SE2 = ee.ImageCollection('COPERNICUS/S2')
```



Hình 3.6 - Lớp phủ rừng ven biển Hải Phòng

Cuối cùng trong phần thiết lập này, chỉ định tập dữ liệu điểm mẫu huấn luyện (training) và xác minh (validtion) với nhãn là rừng hoặc không phải là rừng.

- Bộ dữ liệu huấn luyện (1000 điểm) gồm:
 - 500 điểm dán nhãn là rừng.
 - 500 điểm dán nhãn không phải là rừng.
- Bộ dữ liệu xác minh (300 điểm) gồm:
 - 150 điểm dán nhãn là rừng.
 - 150 điểm dán nhãn không phải là rừng.

```
TRAIN_HAS_FOREST = ee.FeatureCollection(
    'users/trhoa06/unet/Points_HaiPhong_HasForest_Train_500')
TRAIN_NON_FOREST = ee.FeatureCollection(
    'users/trhoa06/unet/Points_HaiPhong_NonForest_Train_500')
VALID_HAS_FOREST = ee.FeatureCollection(
    'users/trhoa06/unet/Points_HaiPhong_HasForest_Valid_150')
VALID_NON_FOREST = ee.FeatureCollection(
    'users/trhoa06/unet/Points_HaiPhong_NonForest_Valid_150')
TRAIN = TRAIN_HAS_FOREST.merge(TRAIN_NON_FOREST)
VALID = VALID_HAS_FOREST.merge(VALID_NON_FOREST)
```

3.3.2 Xây dựng các hàm xử lý.

3.3.2.1 Hàm loại bỏ mây & dải mây mỏng.

❖ Hàm loại bỏ mây

Hàm sau đây cho phép tạo mặt nạ mây bằng cách sử dụng kênh QA trong ảnh Sentinel-2. Tham số đầu vào là ảnh Sentinel-2¹⁷, kết quả trả về là ảnh Sentinel-2 đã được loại bỏ mây. (Google Earth Engine Developers)

```
# Hàm loại bỏ mây ảnh sentinel 2
def maskS2clouds (img):
    qa = img.select('QA60')
    # Bits 10 and 11 are clouds and cirrus, respectively
    cloudBitMask = (1 << 10)
    cirrusBitMask = (1 << 11)
    # Both flags should be set to zero, indicating clear conditions.
    mask = qa.bitwiseAnd(cloudBitMask).eq(0)\
        .And(qa.bitwiseAnd(cirrusBitMask).eq(0))
    return img.updateMask(mask)\
        .divide(10000)\
        .copyProperties(img, ["system:time_start"])
```

❖ Hàm loại bỏ dải mây mỏng.

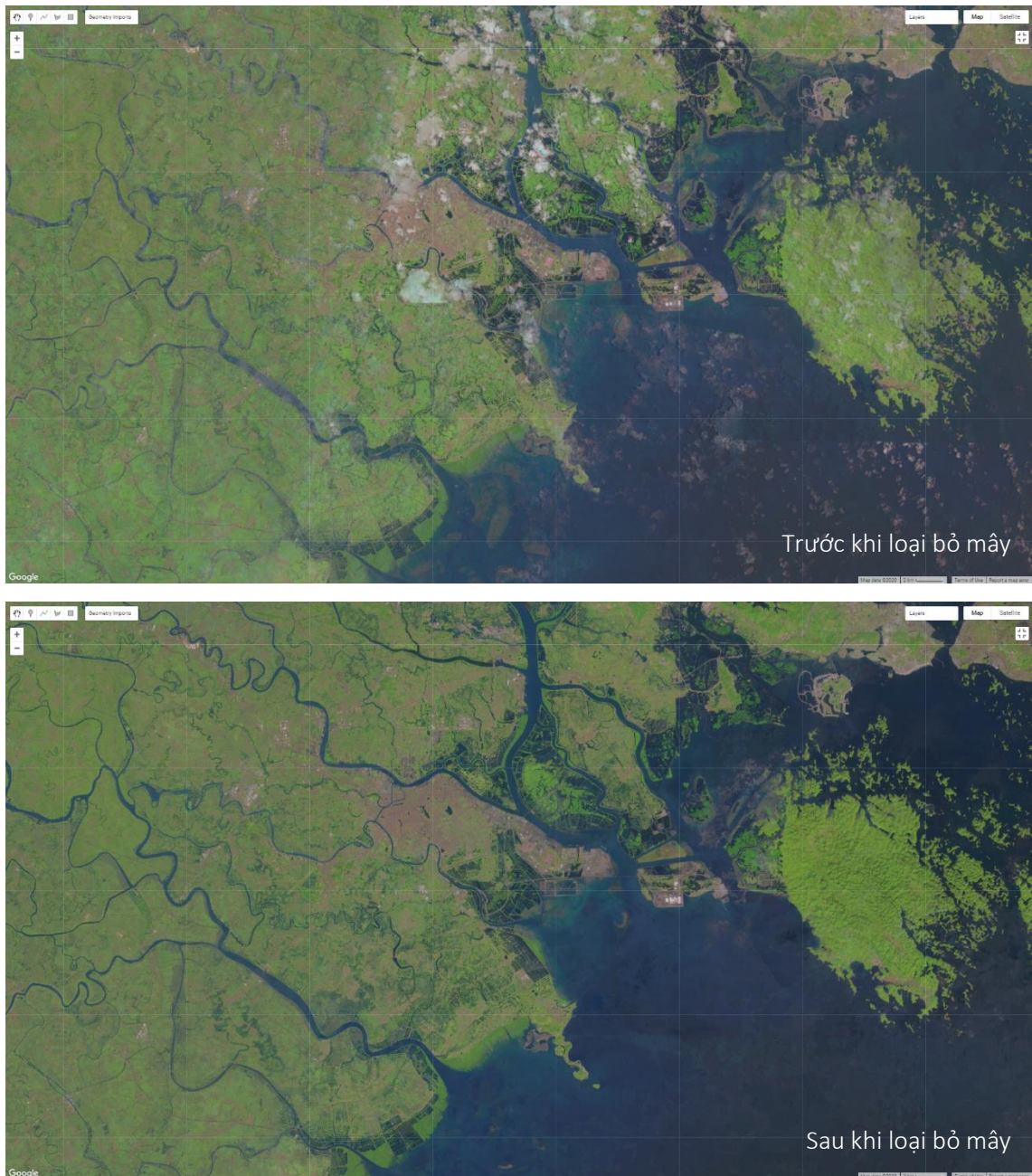
Độ che phủ của đám mây là một trong những mối quan tâm lớn nhất trong viễn thám quang học, bởi vì nó cản trở việc giám sát liên tục bề mặt Trái đất [15]. Hàm này giúp tăng cường chất lượng ảnh đầu ra, giảm sai lệch do ảnh hưởng của dải mây.

```
def rescale(img, exp, thresholds):
    return img.expression(exp, {'img': img})\
        .unitscale(thresholds[0], thresholds[1])
def sentinelCloudScore(img):
    toa = img.select(['^(B|QA60).*']).divide(10000)
    score = ee.Image(1)
    # Clouds are reasonably bright in the blue and cirrus bands.
    score = score.min(rescale(toa, 'img.B2', [0.1, 0.5]))
    score = score.min(rescale(toa, 'img.B1', [0.1, 0.3]))
    score = score.min(rescale(toa, 'img.B1 + img.B10', [0.15, 0.2]))
    # Clouds are reasonably bright in all visible bands.
    score = score.min(rescale(toa, 'img.B4 + img.B3 + img.B2', [0.2, 0.8]))
    # Clouds are moist
    ndmi = img.normalizedDifference(['B8', 'B11'])
    score = score.min(rescale(ndmi, 'img', [-0.1, 0.1]))
    # However, clouds are not snow.
    ndsi = img.normalizedDifference(['B3', 'B11'])
    score = score.min(rescale(ndsi, 'img', [0.8, 0.6]))

    score = score.multiply(100).byte();
    return img.addBands(score.rename('score_x'));
```

¹⁷ https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S2

Kết quả xử lý ảnh trước và sau khi thực hiện loại bỏ mây



Hình 3.7 - Kết quả xử lý ảnh loại bỏ mây

3.3.2.2 Hàm tính toán chỉ số từ ảnh vệ tinh.

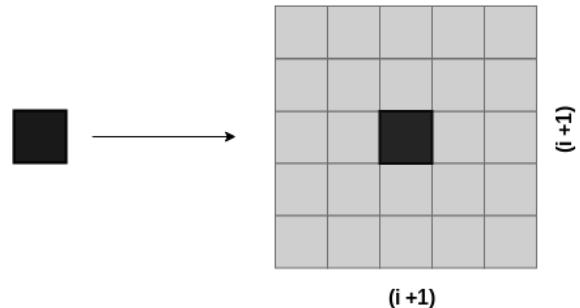
Hàm này cho phép tính toán các chỉ số cần thiết sử dụng, bao gồm: chỉ số thực vật khác biệt chuẩn (NDVI), chỉ số đất khác biệt chuẩn (NDSI) và chỉ số mặt nước khác biệt chuẩn (NDWI).

```
def calcIndexNDs(img):  
    ndvi = img.normalizedDifference(['B8', 'B4']).rename('NDVI')  
    ndsi = img.normalizedDifference(['B3', 'B11']).rename('NDSI')  
    ndwi = img.normalizedDifference(['B8', 'B11']).rename('NDWI');  
    return img.addBands(ndvi).addBands(ndsi).addBands(ndwi)
```

3.3.2.3 Xây dựng bộ dữ liệu huấn luyện.

Để chuẩn bị cho quá trình huấn luyện mạng U-Net, nội dung này sẽ thực hiện việc xây dựng bộ dữ liệu đầy đủ từ các dữ liệu đầu vào gồm: ảnh vệ tinh đã qua tiền xử lý, lớp phủ rừng ven biển và dữ liệu điểm mẫu.

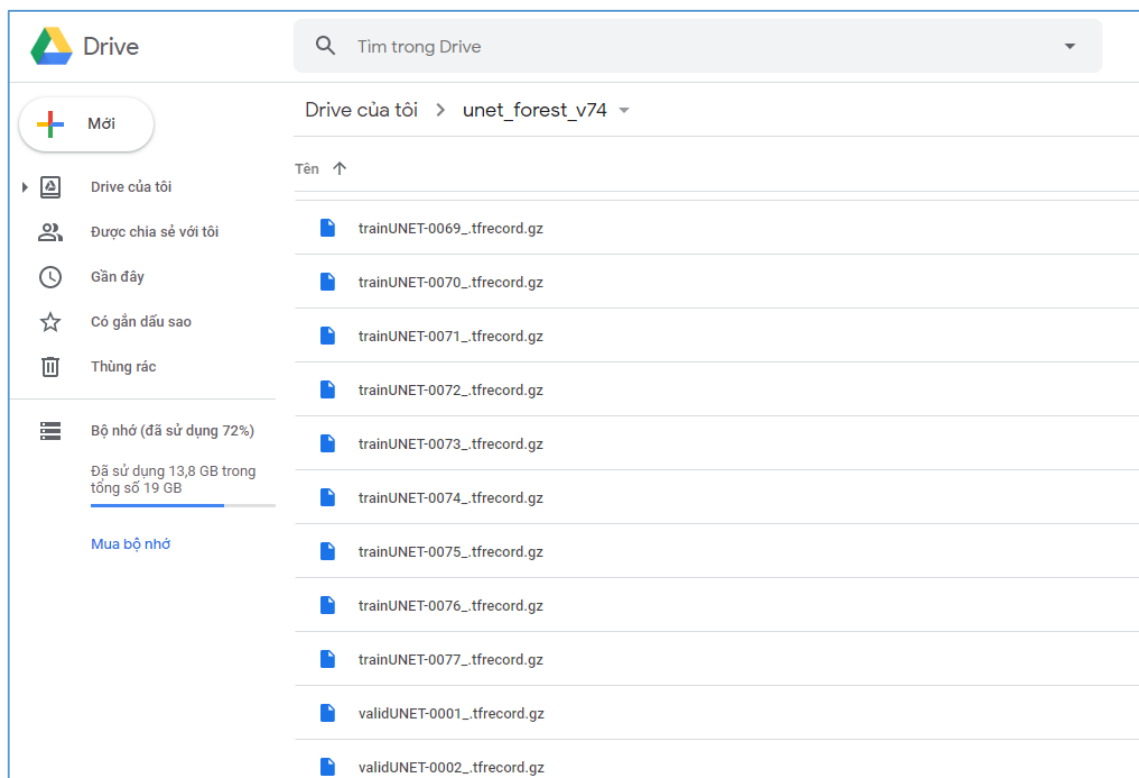
Hàm lấy mẫu dữ liệu gọi và kế thừa hàm `ee.Image.neighborhoodToArray` trong GEE để xuất vùng lân cận của điểm ảnh. `ee.Image.neighborhoodToArray` biến vùng lân cận của từng điểm ảnh (pixel) trong ảnh vô hướng thành ma trận 2D. Trục 0 và 1 của ma trận đầu ra tương ứng với trục Y và X của hình ảnh. Ảnh đầu ra sẽ có nhiều kênh ảnh như đầu vào; mỗi kênh ảnh đầu ra có cùng mặt nạ với kênh ảnh đầu vào tương ứng. Dấu chân và siêu dữ liệu của hình ảnh đầu vào được giữ nguyên.



Hình 3.8 - Xuất vùng lân cận điểm ảnh.

Do số lượng đặc trưng tối đa có thể lưu bị hạn chế. Nên hàm `save_cnn_batch` được xây dựng để lưu bộ dữ liệu đặc trưng thành các phân đoạn phù hợp. Mỗi phân đoạn được đóng gói thành tập tin định dạng `.tfrecord` và lưu trữ.

```
train_filenames = save_cnn_batch(img_src, TRAIN, 'trainUNET', fcp_trains)
valid_filenames = save_cnn_batch(img_src, VALID, 'validUNET', fcp_valids)
```



Hình 3.9 - Dữ liệu huấn luyện được lưu với định dạng `.tfrecord` trong Google Drive.

Tiếp theo, đọc dữ liệu từ tập tin tfrecord đưa vào tf.data.Dataset. Đây là bước tiền xử lý dữ liệu từ các tập tin tfrecord để chuyển đổi thành định dạng phù hợp cho đầu vào mô hình CNN.

```
def input_fn(fileNames,numEpochs=None,shuffle=True,batchSize=16,side=257):
    dataset = tf.data.TFRecordDataset(fileNames, compression_type='GZIP')
    feature_columns = {
        'SWIR2' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'SWIR1' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'RE4' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'NIR' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'RE3' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'RE2' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'RE1' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'R' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'G' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'B' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'NDVI' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'NDSI' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'NDMI' : tf.io.FixedLenFeature([side,side], dtype=tf.float32),
        'forest': tf.io.FixedLenFeature([side, side], dtype=tf.float32)
    }

    # Thực hiện một chức năng phân tích cú pháp
    def parse(example_proto):
        parsed_features = tf.io.parse_single_example(
            example_proto,
            feature_columns)
        parsed_features = {
            key:value[1:side,1:side] for key,
            value in parsed_features.items()
        }
        labels = parsed_features.pop('forest')
        return parsed_features, tf.cast(labels, tf.float32)

    def stack_images(features,label):
        nfeat = tf.transpose(tf.squeeze(tf.stack(list(features.values()))))
        nlabel = (tf.transpose(label))[:, :, tf.newaxis]
        print (f'label: {label} - nfeat: {nfeat} - nlabel: {nlabel}')
        return nfeat, nlabel

    dataset = dataset.map(parse, num_parallel_calls=5)
    dataset = dataset.map(stack_images, num_parallel_calls=5)
    if shuffle:
        dataset = dataset.shuffle(buffer_size = batchSize * 10)
    dataset = dataset.batch(batchSize)
    dataset = dataset.repeat(numEpochs)

    return dataset
```

3.3.3 Thực thi mạng U-Net.

Trong TensorFlow đã tối ưu hóa các thuật toán và đặc biệt cũng cấp sẵn `tf.keras` (đây là một API cấp cao để hỗ trợ xây dựng và huấn luyện các mô hình học sâu). Keras API giúp triển khai nhanh mô hình huấn luyện mà không cần phải xây dựng lại từ đầu.

Tại đây, để xây dựng mạng U-Net với Keras. Mô hình mạng nơ-ron tích chập (CNN) được tạo gồm:

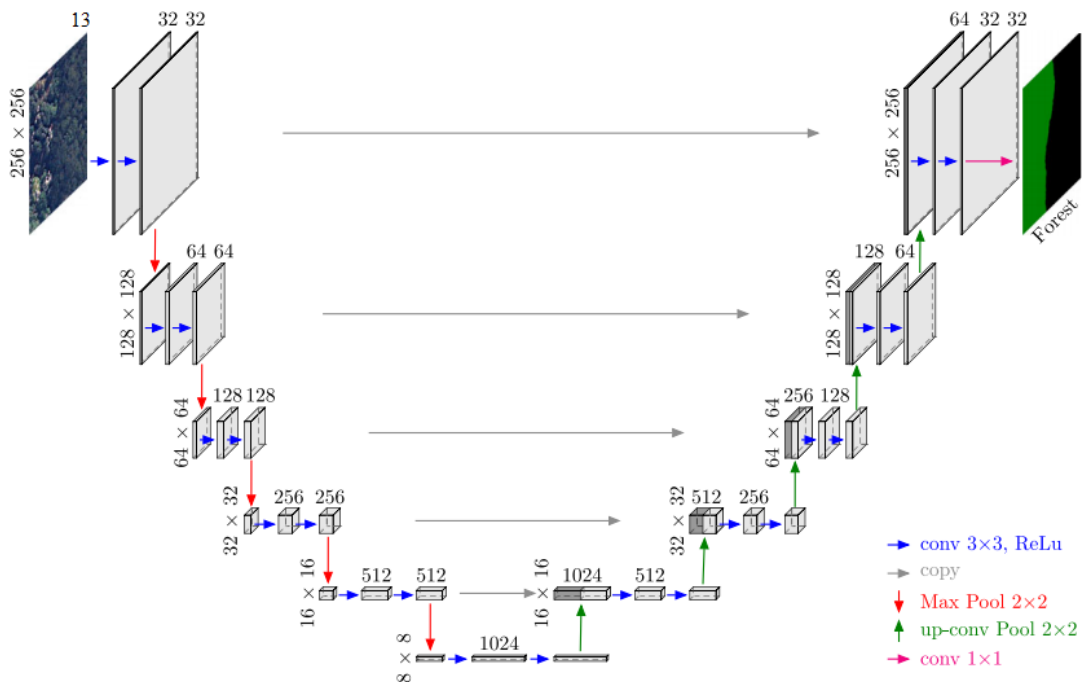
- 5 tầng mã hóa (encoder layers)
- 5 tầng giải mã (decoder layer).
- 1 tầng đầu ra (output layer).

Tầng mã hóa: bao gồm một ngăn xếp tuyến tính gồm các hoạt động Conv, Batchnorm và Relu theo sau là Maxpool. Mỗi MaxPool sẽ giảm độ phân giải không gian của bản đồ tính năng của tôi xuống 2 lần.

Tầng giải mã: bao gồm UpSampling2D, Conv, BatchNorm và ReLU. Sau đó ghép bản đồ tính năng có cùng kích thước ở phía bộ giải mã. Cuối cùng, tôi thêm một hoạt động Conv cuối cùng thực hiện tích chập dọc theo các kênh cho từng điểm ảnh riêng lẻ (kernel_size (1, 1)) tạo ra mặt nạ phân đoạn cuối cùng.

Tầng đầu ra: được tạo thành từ hoạt động Conv cuối cùng để ánh xạ đặc trưng.

Các bước để xây dựng mạng U-Net được giải thích trong hình



Hình 3.10 – Mô hình mạng U-Net thử nghiệm.

3.3.3.1 Định nghĩa mô hình.

Cài đặt mạng U-Net với Keras bằng các tham số như sau:

```
def conv_block(input_tensor, num_filters):
    encoder = layers.Conv2D(num_filters, (3, 3), padding='same')(input_tensor)
    encoder = layers.BatchNormalization()(encoder)
    encoder = layers.Activation('relu')(encoder)
    encoder = layers.Conv2D(num_filters, (3, 3), padding='same')(encoder)
    encoder = layers.BatchNormalization()(encoder)
    encoder = layers.Activation('relu')(encoder)
    return encoder

def encoder_block(input_tensor, num_filters):
    encoder = conv_block(input_tensor, num_filters)
    encoder_pool = layers.MaxPooling2D((2, 2), strides=(2, 2))(encoder)
    return encoder_pool, encoder

def decoder_block(input_tensor, concat_tensor, num_filters):
    decoder = layers.Conv2DTranspose(num_filters, (2, 2), strides=(2, 2), padding='same')(input_tensor)
    decoder = layers.concatenate([concat_tensor, decoder], axis=-1)
    decoder = layers.BatchNormalization()(decoder)
    decoder = layers.Activation('relu')(decoder)
    decoder = layers.Conv2D(num_filters, (3, 3), padding='same')(decoder)
    decoder = layers.BatchNormalization()(decoder)
    decoder = layers.Activation('relu')(decoder)
    decoder = layers.Conv2D(num_filters, (3, 3), padding='same')(decoder)
    decoder = layers.BatchNormalization()(decoder)
    decoder = layers.Activation('relu')(decoder)
    return decoder

inputs = layers.Input(shape=IMG_SHAPE) # 256
encoder0_pool, encoder0 = encoder_block(inputs, 32) # 128
encoder1_pool, encoder1 = encoder_block(encoder0_pool, 64) # 64
encoder2_pool, encoder2 = encoder_block(encoder1_pool, 128) # 32
encoder3_pool, encoder3 = encoder_block(encoder2_pool, 256) # 16
encoder4_pool, encoder4 = encoder_block(encoder3_pool, 512) # 8
center = conv_block(encoder4_pool, 1024)
decoder4 = decoder_block(center, encoder4, 512) # 16
decoder3 = decoder_block(decoder4, encoder3, 256) # 32
decoder2 = decoder_block(decoder3, encoder2, 128) # 64
decoder1 = decoder_block(decoder2, encoder1, 64) # 128
decoder0 = decoder_block(decoder1, encoder0, 32) # 256
outputs = layers.Conv2D(1, (1, 1), activation='sigmoid')(decoder0)

model = models.Model(inputs=[inputs], outputs=[outputs])
model.summary()
```

Kết quả xây dựng thu được:

- Tổng số tham số: 31,129,377
- Số tham số có thể huấn luyện: 31,113,377
- Số tham số không thể huấn luyện: 16,000

Xem thêm hoạt động đầy đủ của mỗi lớp trong mạng U-Net tại phụ lục 2

3.3.3.2 Định nghĩa hàm mất mát.

Mục đích của các hàm mất mát (loss functions) là để giảm đi sai số trong việc dự đoán dữ liệu đầu ra. Xác định hàm mất và trọng số tùy chỉnh là dễ dàng với Keras. Chỉ cần định nghĩa một hàm lấy cả nhãn thật cho một ví dụ cụ thể và nhãn dự đoán cho cùng một ví dụ đưa ra tương tự.

Ở đây tôi sử dụng `dice_loss` vì nó hoạt động tốt khi phát sinh lỗi mất cân bằng do thiết kế và sử dụng `binary_crossentropy` để tính toán tổn thất `cross_entropy` giữa nhãn thật và nhãn dự đoán.

```
from tensorflow.keras import losses

def dice_coeff(y_true, y_pred):
    smooth = 1.
    # Flatten
    y_true_f = tf.reshape(y_true, [-1])
    y_pred_f = tf.reshape(y_pred, [-1])
    intersection = tf.reduce_sum(y_true_f * y_pred_f)
    score = (2. * intersection + smooth) / (tf.reduce_sum(y_true_f) + tf.reduce_sum(y_pred_f) + smooth)
    return score

def dice_loss(y_true, y_pred):
    loss = 1 - dice_coeff(y_true, y_pred)
    return loss

def bce_dice_loss(y_true, y_pred):
    loss = losses.binary_crossentropy(y_true, y_pred) + dice_loss(y_true, y_pred)
    return loss
```

3.3.3.3 Hiệu chỉnh trọng số.

Tôi sử dụng hàm mất mát tùy chỉnh vừa xây dựng để giảm thiểu sự mất mát. Ngoài ra, tôi chỉ định những trọng số (metrics) mình muốn theo dõi khi thực hiện huấn luyện huấn luyện.

Thuật toán tối ưu hóa Adam là một phần mở rộng cho việc giảm độ dốc ngẫu nhiên. Adam được biết là hội tụ nhanh trong quá trình huấn luyện và được áp dụng rộng rãi hơn cho các ứng dụng học sâu trong thị giác máy tính và xử lý ngôn ngữ tự nhiên. Kết quả thực nghiệm cho thấy Adam làm việc tốt hơn so với các phương pháp tối ưu hóa khác.

```
from tensorflow.keras.utils import plot_model
model.compile(optimizer='adam',
              loss=bce_dice_loss,
              metrics=[dice_loss, 'accuracy'])
plot_model(model)
```

3.3.4 Huấn luyện mô hình.

Huấn luyện mô hình với `tf.data` chỉ đơn giản là cung cấp chức năng `fit` của mô hình với tập dữ liệu huấn luyện/xác minh, số bước (step) và số lần duyệt (epoch).

Quá trình huấn luyện dữ liệu gọi lại `ModelCheckpoint` để lưu mô hình vào đĩa sau mỗi lần duyệt (epoch). Tôi cấu hình nó sao cho nó chỉ lưu mô hình hiệu suất cao nhất của tôi. Mặc định, nó lưu kiến trúc mô hình, các trọng số, cũng như thông tin về quy trình huấn luyện như trạng thái của trình tối ưu hóa..vv.

Dưới đây là đoạn mã thực hiện huấn luyện mô hình.

```
from tensorflow import keras
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
import os
import datetime

logdir = os.path.join("logs/fit/",
                      datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)
es = EarlyStopping(monitor='val_loss', patience=10)
mcp = ModelCheckpoint(filepath='forest_model.h5',
                      monitor='val_loss',
                      save_best_only=True)

N_TRAIN = TRAIN.size().getInfo()
N_VALID = VALID.size().getInfo()
BATCH_SIZE = 10

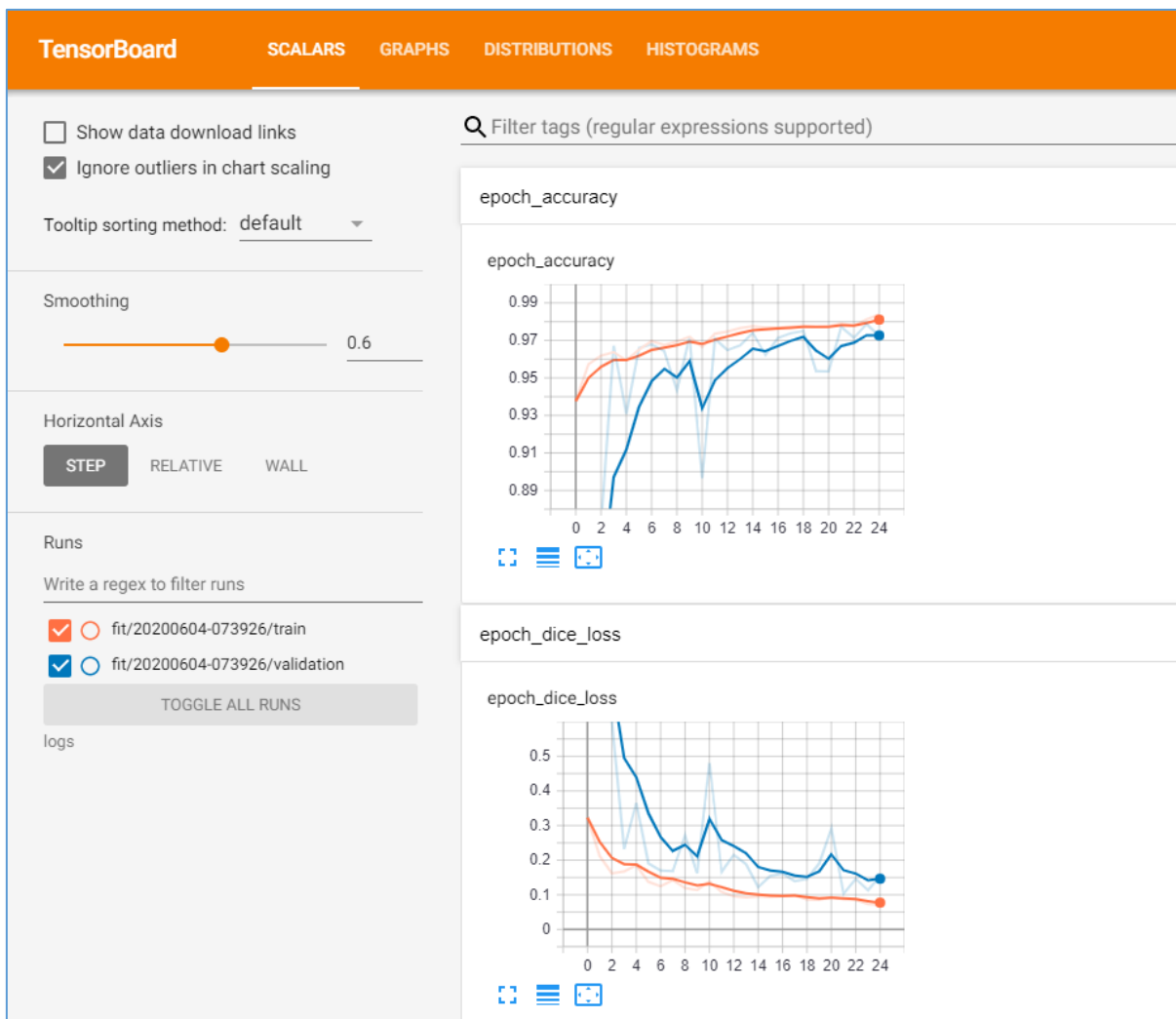
history = model.fit(x=train_db,
                    epochs=EPOCHS,
                    steps_per_epoch= int(np.ceil(N_TRAIN/float(BATCH_SIZE))),
                    validation_steps= int(np.ceil(N_VALID/float(BATCH_SIZE))),
                    validation_data=valid_db,
                    callbacks=[tensorboard_callback, es, mcp])
```

Thông tin	TPU
Huấn luyện (Training)	1000 điểm
Xác minh (Validation)	300 điểm
Số lần duyệt (Epochs)	25
Thời gian duyệt trung bình (s)	123s
Tổng thời gian (s)	1h

Bảng 3.3 - Thông tin huấn luyện mô hình.

Kết quả huấn luyện thu được mô hình tối ưu đáp ứng mục tiêu yêu cầu, mô hình có thể được sử dụng trong tương lai mà không cần phải huấn luyện lại. Mô hình forest_unet.h5 có thể sử dụng dự đoán rừng từ ảnh Sentinel-2 bất kỳ. Kết quả mô hình được huấn luyện được công khai trong liên kết sau: https://bit.ly/unet_model.

Mô hình kết quả có độ chính xác huấn luyện gần bằng 0,9838 và xác minh gần bằng 0,9803. Chúng ta có thể hình dung biểu đồ huấn luyện trong thư viện TensorBoard.



Hình 3.11 - Biểu đồ đường cong quá trình huấn luyện.

Giải thích: Từ biểu đồ thấy rằng hàm mất giảm, nó cho thấy rằng các tham số đang học. Nhưng tuy nhiên mô hình có thể được huấn luyện nhiều hơn nữa. Lý tưởng nhất là độ chính xác tăng và tổn thất giảm theo thời gian, các overfitting xảy ra chủ yếu khi có rất ít dữ liệu và tôi buộc phải cung cấp thêm nhiều dữ liệu hơn.

Mô hình kết quả được công khai: https://bit.ly/cnn_forest_change_detection

3.3.5 Dự đoán.

Các bước thực hiện dự đoán diện tích rừng từ ảnh vệ tinh, gồm:

- Định nghĩa hàm dự đoán.
- Xác định phạm vi vùng dự đoán.
- Lấy ảnh theo phạm vi vùng dự đoán.
- Tạo bộ dữ liệu dự đoán (tfrecord) từ ảnh dự đoán
- Dự đoán.
- Hiển thị kết quả.

Dưới đây là các đoạn mã thực thi:

```
# Hàm dự đoán ảnh
def predict_input_fn(fileNames, side, bands):
    dataset = tf.data.TFRecordDataset(fileNames, compression_type='GZIP')

    featuresDict = {x:tf.io.FixedLenFeature([side, side], dtype=tf.float32)
                    for x in bands}
def parse_image(example_proto):
    parsed_features = tf.io.parse_single_example(example_proto, featuresDict)
    return parsed_features

def stack_images(features):
    nfeat = tf.transpose(tf.squeeze(tf.stack(list(features.values()))))
    return nfeat

dataset = dataset.map(parse_image, num_parallel_calls=5)
dataset = dataset.map(stack_images, num_parallel_calls=5)
dataset = dataset.batch(side*side)
return dataset

# Phạm vi dự đoán
region_predict = ee.FeatureCollection(
    [ee.Feature(
        ee.Geometry.Polygon(
            [[[106.65038471541244, 20.576500186981928],
             [106.72351246199447, 20.663901372758993],
             [106.76951771101791, 20.78078606773581],
             [106.77089100203354, 20.74033619189288],
             [106.5981996568187, 20.62534821063121]]]]))])

# Lấy ảnh sentinel-2
img_predict = SE2\
    .filterBounds(region_predict)\
    .filterDate(START_PRED, END_PRED)\
    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', CLOUD_PERCENT))\
    .map(maskS2clouds)\
    .map(sentinelCloudScore)\
    .map(calcIndexNDs)\
    .median()\
    .clip(region_predict)
```

```

# Tạo bộ dữ liệu dự đoán.
imageTask = ee.batch.Export.image.toDrive(
    image=img_des,
    description='Image Export',
    fileNamePrefix=imageFilePrefix,
    folder=OUTPUT_UNET,
    scale=10,
    fileFormat='TFRecord',
    region=region_predict.geometry().getInfo()['coordinates'],
    formatOptions=imageExportFormatOptions,
)

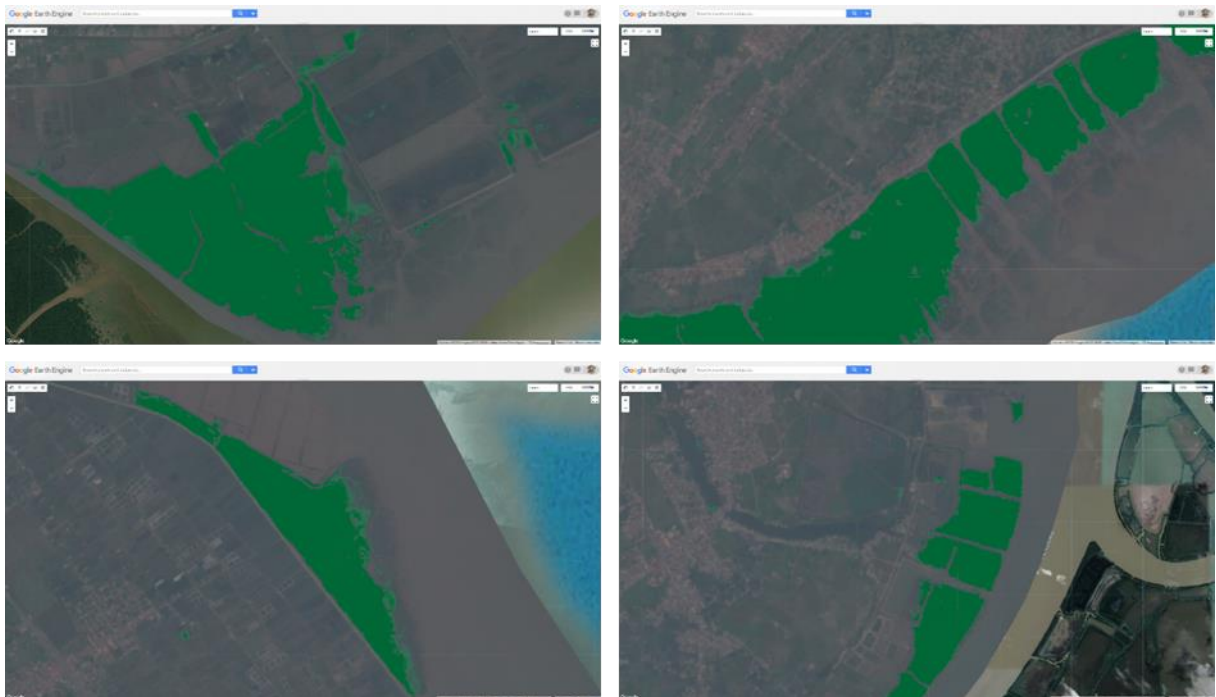
# Dự đoán.
predict_db = predict_input_fn(fileNames=imageFilesList,
                              side=256,
                              bands=RENAME_BANDS)
predictions = model.predict(predict_db)

# Hiển thị kết quả.
!earthengine --no-use_cloud_api upload image
--asset_id={output_assetID} {output_imagefile} {jsonfile}

img_result = ee.Image(path_image_result)
forest_area = img_result.updateMask(img_result).getMapId()
folium.TileLayer(
    tiles=forest_area['tile_fetcher'].url_format,
    attr='Map Data &copy; <a href="https://earthengine.google.com/">Google Earth Engine
    overlay=True,
    name='forest_area',
).add_to(map)

```

Một số kết quả dự đoán thử nghiệm:



Hình 3.12 - Kết quả dự đoán ảnh thử nghiệm.

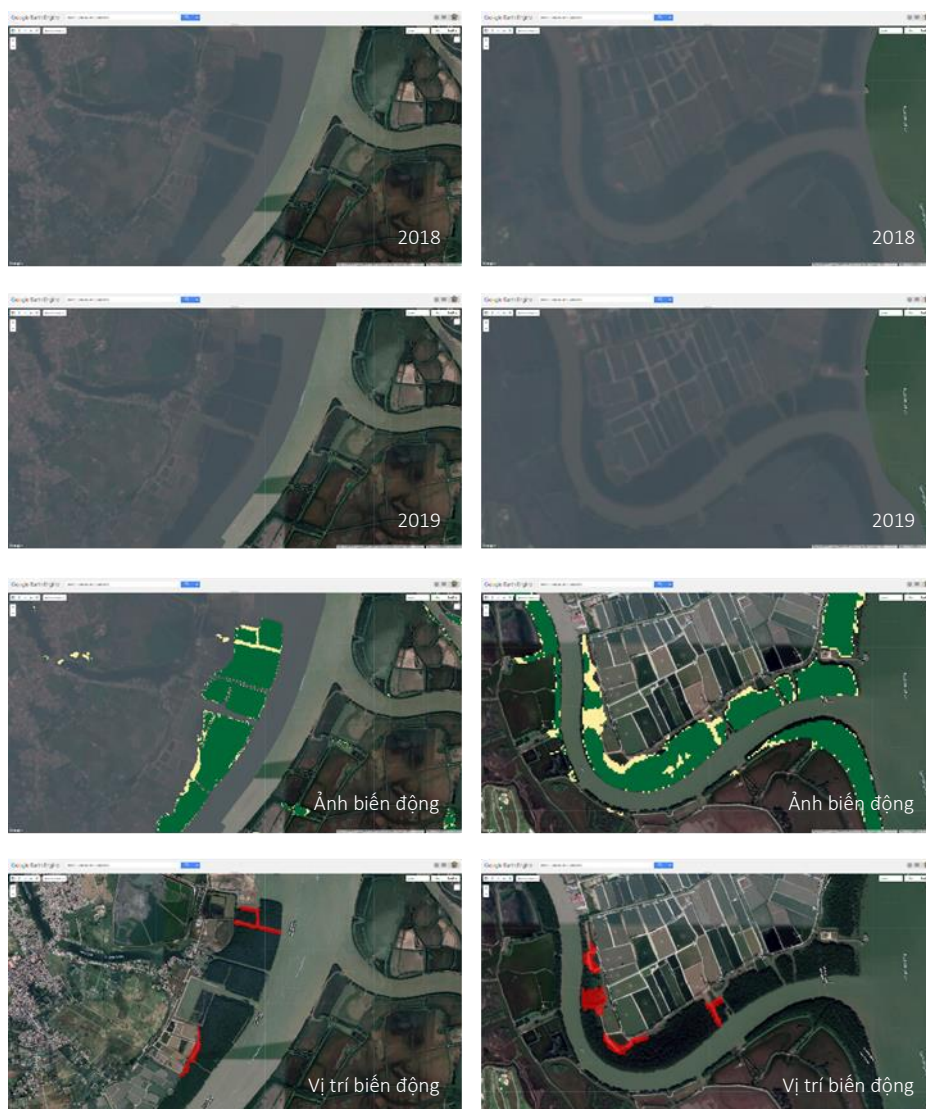
3.4 Kết quả thử nghiệm và đánh giá.

3.4.1 Kết quả thử nghiệm phương pháp U-Net phát hiện biến động rừng.

Sử dụng GIS hoặc GEE chồng xếp, tính toán, phân tích và so sánh giữa lớp dữ liệu hiện trạng rừng công bố hàng năm với kết quả lớp phủ rừng thu được từ dự đoán ảnh vệ tinh dựa trên mạng U-Net nhằm phát hiện ra các lô rừng biến động. Lô rừng biến động là các lô có diện tích liên vùng lớn hơn 0,3 ha.

Lớp phủ rừng thu được từ dự đoán ảnh vệ tinh tại mỗi giai đoạn là thông tin quan trọng để dự báo biến động rừng. Có thể sử dụng đơn lớp phủ rừng hoặc đa lớp phủ rừng ở nhiều giai đoạn thời gian khác tại một vị trí để tăng độ tin cậy cho kết quả dự báo biến động rừng.

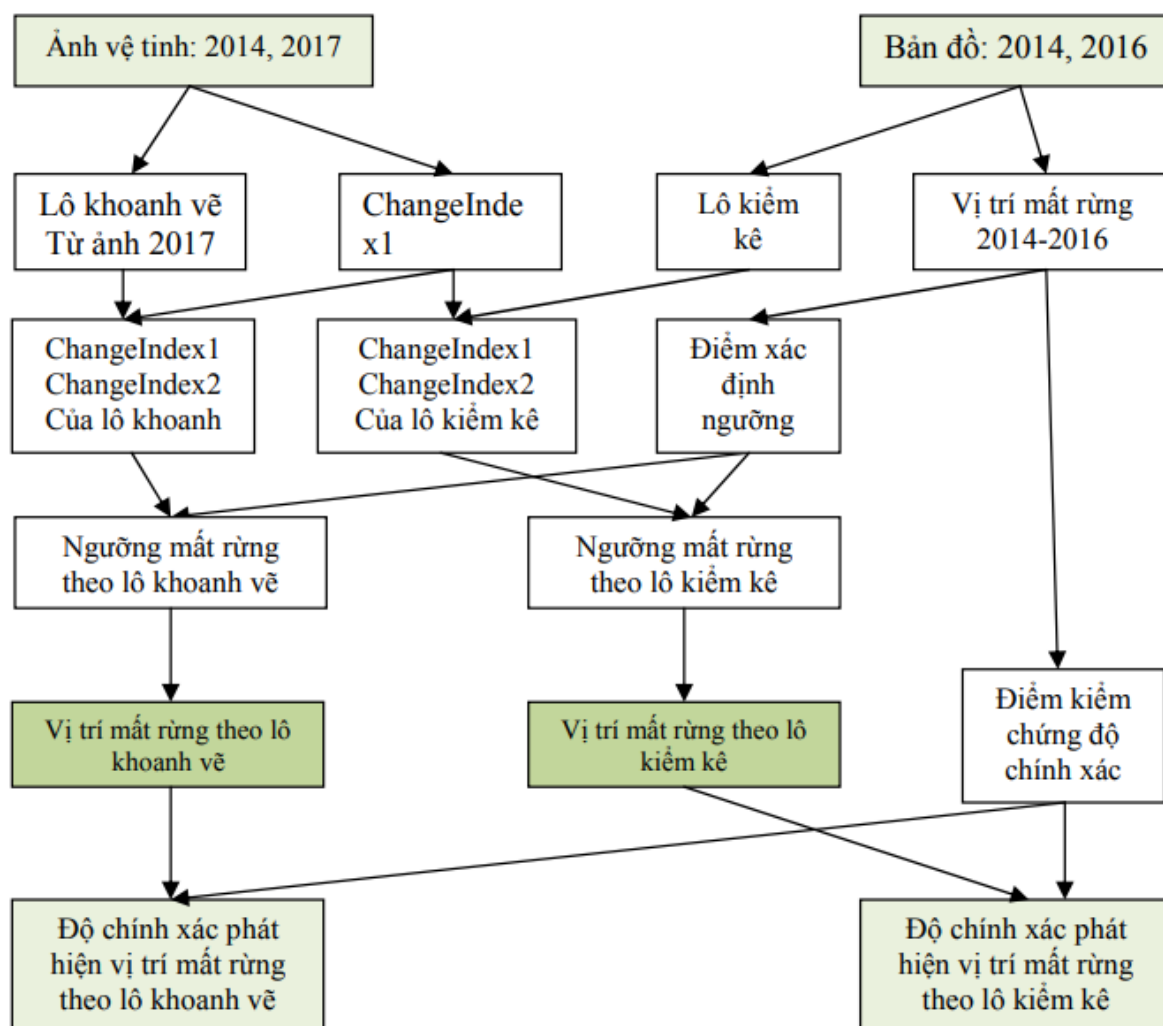
Dưới đây là một số hình ảnh kết quả phát hiện mất rừng ở những vị trí khác nhau tại Thành phố Hải Phòng.



Hình 3.13 - Kết quả thử nghiệm, phát hiện biến động mất rừng tại Huyện Thủy Nguyên, TP. Hải Phòng

3.4.2 So sánh kết quả phát hiện biến động giữa phương pháp MCVA và U-Net.

MCVA (viết tắt của Multi-variant Change Vector Analysis) là phương pháp xác định vị trí mất rừng bằng phương pháp phân tích véc tơ thay đổi đa biến được nhóm tác giả công bố năm 2017. MCVA dựa trên tư liệu ảnh viễn thám miễn phí để phát hiện vị trí mất rừng. Phương pháp phân tích véc tơ thay đổi đa biến dựa trên 2 chiều chính là: (1) chiều thay đổi chỉ số thực vật khác biệt chuẩn (NDVI) và (2) chiều thay đổi chỉ số đất khác biệt chuẩn (NDSI).



Hình 3.14 - Sơ đồ quá trình xử lý, phân tích dữ liệu.

Độ chính xác theo công bố của nghiên cứu có thể đạt được từ 91,6% đến 99,2% [2]. Trong năm 2018-2019 nhóm tác giả cũng đã áp dụng phương pháp MCVA xây dựng công cụ giám sát rừng, kết quả được công bố tại địa chỉ rungvenbien.ifee.edu.vn

Đây thực sự là kênh dữ liệu tốt để tôi có thể kiểm định phương pháp học máy vừa xây dựng cũng như đánh giá tiềm năng và sự hiệu quả của ứng dụng mạng nơ-ron trong giám sát tài nguyên rừng.

❖ Kịch bản so sánh ở cùng vị trí địa lý giữa phương pháp MCVA và U-Net.

Để kiểm chứng mô hình và kết quả xác định biến động giữa hai phương pháp MCVA và U-Net. Trong nghiên cứu này xác định một số quy tắc chung thực hiện:

- Nguồn dữ liệu thử nghiệm: Ảnh vệ tinh Sentinel-2.
- Dữ liệu trạng thái rừng: Hiện trạng rừng TP. Hải Phòng năm 2018.
- Vị trí thử nghiệm: Thành phố Hải Phòng.
- Thời gian thử nghiệm: Từ 01/2018 đến 12/2019.

Kịch bản thực hiện:

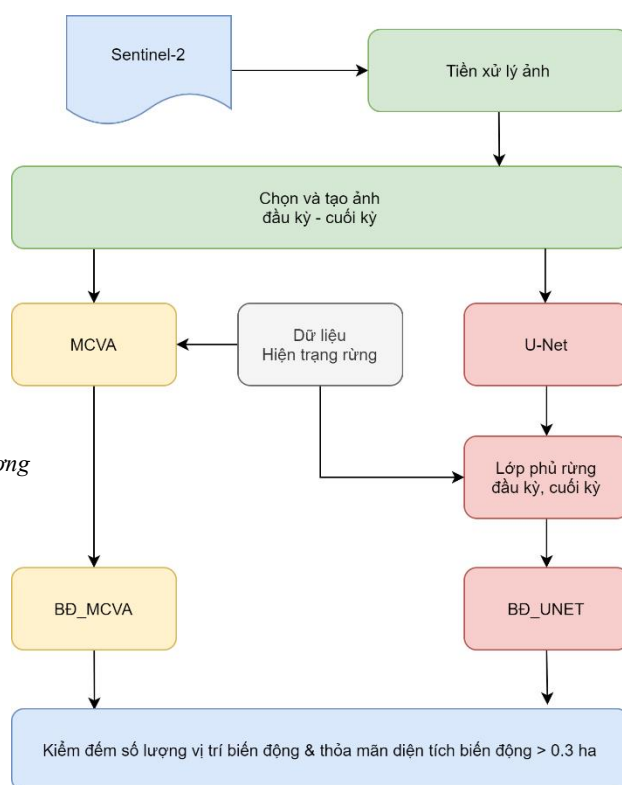
Bước 1: Tiền xử lý ảnh Sentinel-2 và tạo dữ liệu đầu vào.

- Ảnh đầu kỳ từ 01/01/2018 đến 31/12/2018
- Ảnh cuối kỳ từ 01/01/2019 đến 31/12/2019.

Bước 2: Sử dụng phương pháp MCVA xác định biến động từ ảnh đầu kỳ, ảnh cuối kỳ và dữ liệu hiện trạng rừng. Kết quả thu được là ảnh biến động BD_MCVA.

Bước 3: Sử dụng phương pháp U-Net dự đoán ảnh đầu vào dựa trên kết quả mô hình đã huấn luyện để xây dựng lớp phủ rừng. Ảnh đầu kỳ sẽ tạo ra lớp phủ rừng 1 (ForestCover1), ảnh cuối kỳ sẽ tạo ra lớp phủ rừng 2 (ForestCover2); kết hợp các lớp phủ rừng với dữ liệu hiện trạng rừng tính toán xác định biến động. Kết quả thu được là ảnh biến động BD_UNET.

Bước 4: Kiểm đếm số lượng vị trí biến động và diện tích biến động phát hiện cùng tại vị trí thử nghiệm từ ảnh BD_MCVA và BD_UNET thỏa mãn tiêu chí diện tích liên vùng 0,3 ha trở lên.

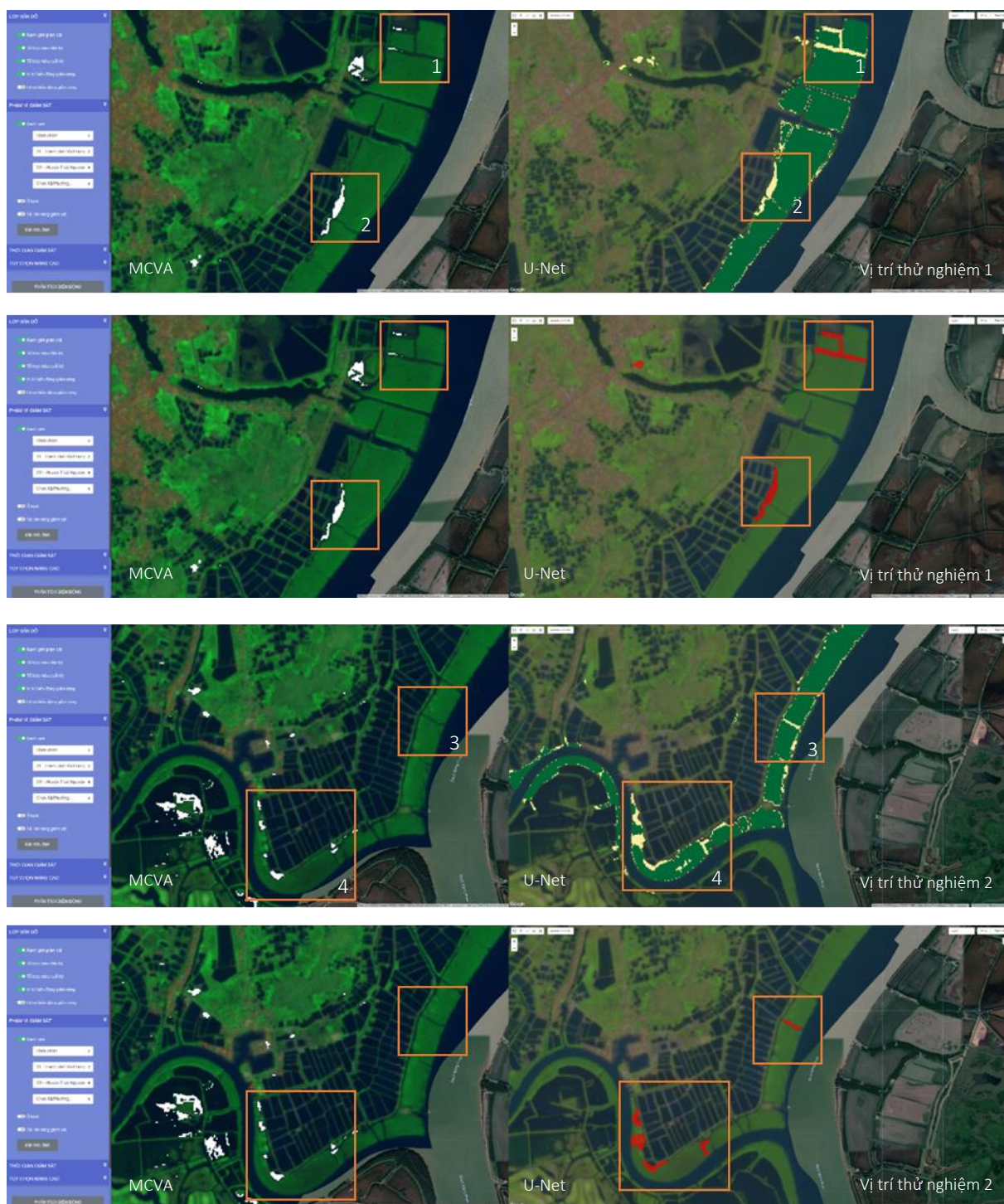


Hình 3.15 - Kịch bản so sánh ở cùng vị trí địa lý giữa phương pháp MCVA và U-Net

❖ **Kết quả so sánh ở cùng vị trí địa lý giữa phương pháp MCVA và U-Net.**

Dưới đây là một số hình ảnh kết quả phát hiện biến động giữa phương pháp MCVA và U-Net.

Phương pháp U-Net xác định chính xác biến động rừng các các vị trí: 1, 2, 3, 4. Phương pháp MCVA chỉ xác định được biến động rừng tại vị trí 2 và 4.



Hình 3.16 - So sánh kết quả phát hiện biến động mất rừng giữa phương pháp MCVA và U-Net.

3.4.3 *Đánh giá kết quả.*

Từ thực nghiệm cho thấy, kết quả phát hiện và xác định vị trí biến động rừng bằng phương pháp U-Net cho thấy mức độ chi tiết biến động cao hơn, phát hiện ra những vị trí biến động mà phương pháp MCVA đã bỏ qua và diện tích các lô biến động gần với kết quả kiểm nghiệm thực địa. Điều này mở ra nhiều hướng nghiên cứu ứng dụng mới rất tiềm năng trong việc giám sát bảo vệ rừng ven biển, bảo vệ hệ sinh thái ven biển.

3.5 Tổng kết chương 3

Trong chương này, luận văn trình bày quá trình hiện thực hóa quy trình giải bài toán phát hiện mất rừng từ ảnh vệ tinh dựa trên mạng nơ-ron đã xây dựng (chương 2). Cụ thể, đã chỉ rõ phần mềm, môi trường, thư viện sử dụng để thử nghiệm; tiếp đó là thực hiện các bước chuẩn bị dữ liệu đầu vào, cài đặt thực nghiệm xây dựng các hàm chức năng tiền xử lý, xây dựng kiến trúc mạng U-Net, huấn luyện và dự đoán. Sau những hiệu chỉnh và thu được mô hình kết quả tối ưu, kết quả biến động thu được từ mạng U-Net được thực nghiệm so sánh với mô hình MCVA để đánh giá tính chính xác và sự vượt trội của mô hình mới. Khẳng định hướng tiếp cận ứng dụng trí tuệ nhân tạo trong việc giám sát bảo vệ tài nguyên rừng nói chung, bảo vệ RVB nói riêng là vô cùng chính xác.

CHƯƠNG 4

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Những đóng góp của luận văn.

Với mục tiêu nghiên cứu bài toán phát hiện biến động mất rừng ven biển từ ảnh vệ tinh dựa trên mạng nơ-ron. Luận văn đã đi sâu nghiên cứu các vấn đề xung quanh của bài toán và kiến trúc mạng U-Net để phân đoạn hình ảnh phát hiện đối tượng rừng:

Nhưng kết quả đóng góp đã đạt được trong luận văn:

- Khảo sát thu thập đầy đủ dữ liệu ảnh vệ tinh và hiện trạng rừng ven biển.
- Chỉ ra các đặc trưng quan trọng của ảnh vệ tinh, khả năng sử dụng ảnh vệ tinh để giám sát phát hiện biến động rừng, đặc biệt là biến động mất rừng.
- Nghiên cứu và ứng dụng thành công các kỹ thuật xử lý ảnh vệ Sentinel-2.
- Nghiên cứu mạng CNN phát hiện và phân đoạn hình ảnh phù hợp bài toán phát hiện biến động mất rừng từ ảnh vệ tinh.
- Cài đặt thử nghiệm, huấn luyện dữ liệu, hiệu chỉnh tham số và dự đoán thành công với mạng U-Net.
- So sánh đánh giá kết quả phát hiện biến động rừng giữa phương pháp MCVA và U-Net vừa xây dựng.

Việc sử dụng dịch vụ điện toán đám mây để khai thác dữ liệu ảnh vệ tinh, tính toán, xử lý vào huấn luyện học máy cho hiệu suất cao và thời gian rút ngắn đi rất nhiều.

Kết quả thực nghiệm cho thấy kiến trúc mạng nơ-ron U-Net dự đoán diện tích RVB thỏa đáng, đạt độ chính xác 98,38%. U-Net có thể có thể dự đoán nhanh ảnh vệ tinh để lập bản đồ lớp phủ rừng ven biển và kiểm nghiệm thực tế cho U-Net có hiệu suất tốt so với phương pháp truyền thống.

4.2 Hướng phát triển.

Tuy đạt được một số kết quả nêu trên, nhưng luận văn còn một số hạn chế do điều kiện về mặt thời gian, dữ liệu thu thập và trình độ của học viên. Vì vậy, hướng nghiên cứu tiếp theo của học viên là:

- Tiếp tục cải tiến bộ dữ liệu huấn luyện để kết quả dự đoán tốt hơn nữa.
- Sử dụng kết hợp nhiều nguồn ảnh vệ tinh để tăng cường thời gian giám sát.
- Sử dụng thêm nhiều nhãn đối tượng rừng, để dự báo chi tiết hơn đối tượng.
- Tăng số lần huấn luyện và sử dụng kết hợp các thuật toán, mô hình thông minh khác để cải thiện và nâng cao độ chính xác của lớp phủ rừng.

KẾT LUẬN

Ứng dụng học máy trong giám sát biến động rừng là một hướng đi đúng đắn và kịp thời. Mô hình kết quả thu được đạt độ chính xác cao, năng lực dự đoán ảnh nhanh, cho kết quả đáp ứng được yêu cầu thực tế.

Nghiên cứu cho thấy việc ứng dụng học máy giải quyết tốt các yêu cầu thực tế đặt ra. Giúp nâng cao hiệu quả trong việc quản lý và giám sát bảo vệ tài nguyên rừng ở Việt Nam.

DANH MỤC CÁC TÀI LIỆU THAM KHẢO

- [1] **Trần Trọng Đức, Phạm Bách Việt.** Giám sát biến động rừng ngập mặn Cần Giờ TP. HCM sử dụng kỹ thuật viễn thám và GIS. *Báo cáo nghiên cứu khoa học cấp Bộ*. 2005.
- [2] **Nguyễn Thanh Hoàn, Phạm Văn Duẩn, Lê Sỹ Doanh, Nguyễn Văn Dũng.** Xác định vị trí mất rừng bằng phương pháp phân tích véc tơ thay đổi đa biến (MCVA) trên tư liệu vệ tinh Landsat-8. *Tạp chí Khoa học và Công nghệ Lâm nghiệp*. 2017.
- [3] **Phạm Văn Duẩn, Lê Sỹ Doanh, Vũ Thị Thìn, Nguyễn Văn Thị, Hoàng Văn Khiên, Phạm Tiến Dũng, Đinh Văn Tuyền.** Đánh giá khả năng khai thác ảnh vệ tinh quang học miễn phí phục vụ giám sát lớp phủ mặt đất tại Việt Nam. *Tạp chí Khoa học và Công nghệ Lâm nghiệp*. 2019.
- [4] **Nguyễn Văn Thị, Trần Quang Bảo, Lê Sỹ Doanh, Phạm Văn Duẩn, Nguyễn Nam Hải, Trần Xuân Hòa.** Nghiên cứu kết hợp ảnh vệ tinh quang học Sentinel-2 và Radar Sentinel-1 trong phát hiện mất rừng ở tỉnh Gia Lai. *Tạp chí Nông nghiệp và Phát triển nông thôn*. 2020.
- [5] **Phạm Việt Hòa.** Ứng dụng công nghệ tích hợp tư liệu viễn thám và hệ thống tin địa lý xác định sự biến động rừng ngập mặn. *Trường đại học Mở - Địa chất*. 2012.
- [6] **Trần Quang Bảo, Lê Sỹ Doanh, Hoàng Thị Hồng.** Sử dụng ảnh Google Earth để xây dựng bản đồ hiện trạng rừng và đánh giá biến động rừng tại Công ty Lâm nghiệp La Ngà, Tỉnh Đồng Nai. *Tạp chí Khoa học và Công nghệ Lâm nghiệp*. 2018.
- [7] **Nguyễn Hải Hòa, Nguyễn Thị Thu Hiền, Lương Thị Thu Trang.** Ứng dụng GIS và ảnh Landsat đa thời gian xây dựng bản đồ biến động diện tích rừng tại xã vùng đệm Xuân Đài và Kim Thượng, Vườn quốc gia Xuân Sơn. *Tạp chí Khoa học Lâm nghiệp*. 2016.
- [8] **Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, Rebecca Moore.** Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*. 2017.
- [9] **Matthew C. Hansen, David P. Roy, Erik Lindquist, Bernard Adusei, Christopher O. Justice, Alice Altstatt.** A method for integrating MODIS and Landsat data for systematic monitoring of forest cover and change in the Congo Basin. *Remote Sensing of Environment*. 2008.

- [10] **Dave Thau, Dominic Andradi-Brown, Luiz Cortinhas, Cesar Diniz, Pedro Souza-Filho, Greg Fiske, Chandra Giri, Liza Goldberg, David Lagomasino, Uday Pimple, Dario Simonetti, Aurelie Shapiro, Xiangming Xiao.** Mangrove Monitoring in Google Earth Engine. *Medium.com*. 2018.
- [11] **Allen Zhao, Shawn Hu, Chenyao Yu.** Classifying Weather, Terrain, and Deforestation of the Amazon using Deep Multi-task Convolutional Neural Nets. *Stanford University*. 2017.
- [12] **Andrei Stoian, Vincent Poulain, Jordi Inglada, Victor Poughon, Dawa Derksen.** Land Cover Maps Production with High Resolution Satellite Image Time Series and Convolutional Neural Networks: Adaptations and Limits for Operational Systems. *Remote Sensing*. 2019.
- [13] **Pablo Pozzobon de BemOrcID, Osmar Abílio de Carvalho Junior, Renato Fontes Guimarães, Roberto Arnaldo Trancoso Gomes.** Change Detection of Deforestation in the Brazilian Amazon Using Landsat Data and Convolutional Neural Networks. *Remote Sensing*. 2020.
- [14] **Salman H. Khan, Xuming He, Fatih Porikli, Mohammed Bennamoun.** Forest Change Detection in Incomplete Satellite Images With Deep Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*. 2017.
- [15] **Michael Schmitt, Lloyd H. Hughes, Chunping Qiu, Xiao Xiang Zhu.** Aggregating cloud-free Sentinel-2 images with Google Earth Engine. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., IV-2/W7*, 145–152. 2019.

PHỤ LỤC 1
DANH SÁCH CẢNH ẢNH SENTINEL-2 SỬ DỤNG HUẤN LUYỆN

STT	Mã ảnh	Tỷ lệ mây	Vệ tinh chụp	Ngày chụp
1	COPERNICUS/S2/20180322T031539_20180322T032727_T48QXH	2.36	Sentinel-2B	2018-03-22
2	COPERNICUS/S2/20180501T031539_20180501T033025_T48QXH	29.76	Sentinel-2B	2018-05-01
3	COPERNICUS/S2/20180516T031541_20180516T032154_T48QXH	3.22	Sentinel-2A	2018-05-16
4	COPERNICUS/S2/20180521T031539_20180521T032337_T48QXH	22.94	Sentinel-2B	2018-05-21
5	COPERNICUS/S2/20180521T031539_20180521T033216_T48QXH	13.58	Sentinel-2B	2018-05-21
6	COPERNICUS/S2/20180521T031539_20180521T033216_T48QXJ	29.72	Sentinel-2B	2018-05-21
7	COPERNICUS/S2/20180526T031541_20180526T032442_T48QXH	11.69	Sentinel-2A	2018-05-26
8	COPERNICUS/S2/20180620T031729_20180620T032517_T48QXJ	8.87	Sentinel-2A	2018-06-20
9	COPERNICUS/S2/20180630T031539_20180630T032902_T48QXJ	17.12	Sentinel-2B	2018-06-30
10	COPERNICUS/S2/20180705T031541_20180705T031931_T48QXH	12.49	Sentinel-2A	2018-07-05
11	COPERNICUS/S2/20180705T031541_20180705T031931_T48QXJ	2.67	Sentinel-2A	2018-07-05
12	COPERNICUS/S2/20180819T031529_20180819T031937_T48QXH	8.84	Sentinel-2B	2018-08-19

13	COPERNICUS/S2/20180819T031529_20180819T033211_T48QXJ	10.31	Sentinel-2B	2018-08-19
14	COPERNICUS/S2/20181003T031541_20181003T032154_T48QXJ	4.57	Sentinel-2A	2018-10-03
15	COPERNICUS/S2/20181008T031619_20181008T032153_T48QXH	13.35	Sentinel-2B	2018-10-08
16	COPERNICUS/S2/20181008T031619_20181008T032153_T48QXJ	5.57	Sentinel-2B	2018-10-08
17	COPERNICUS/S2/20181013T031651_20181013T032303_T48QXJ	17.26	Sentinel-2A	2018-10-13
18	COPERNICUS/S2/20181028T031829_20181028T032856_T48QXH	15.18	Sentinel-2B	2018-10-28
19	COPERNICUS/S2/20181028T031829_20181028T032856_T48QXJ	28.80	Sentinel-2B	2018-10-28
20	COPERNICUS/S2/20181102T031901_20181102T032732_T48QXH	0.00	Sentinel-2A	2018-11-02
21	COPERNICUS/S2/20181102T031901_20181102T032732_T48QXJ	1.24	Sentinel-2A	2018-11-02
22	COPERNICUS/S2/20181107T031929_20181107T033111_T48QXH	25.02	Sentinel-2B	2018-11-07
23	COPERNICUS/S2/20181107T031929_20181107T033111_T48QXJ	28.76	Sentinel-2B	2018-11-07
24	COPERNICUS/S2/20181127T032049_20181127T032257_T48QXJ	7.84	Sentinel-2B	2018-11-27
25	COPERNICUS/S2/20181217T032129_20181217T032130_T48QXH	24.09	Sentinel-2B	2018-12-17
26	COPERNICUS/S2/20181217T032129_20181217T032130_T48QXJ	0.01	Sentinel-2A	2018-12-17

PHỤ LỤC 2

HOẠT ĐỘNG ĐẦY ĐỦ CỦA CÁC LỚP TRONG MẠNG U-NET

Layer	Type	Output Shape	Param	Connected to
input_1	(InputLayer)	[(None, 256, 256, 13	0	
conv2d	(Conv2D)	(None, 256, 256, 32)	3776	input_1[0][0]
batch_normalization	(BatchNorma	(None, 256, 256, 32)	128	conv2d[0][0]
Activation	(Activation)	(None, 256, 256, 32)	0	batch_normalization[0][0]
conv2d_1	(Conv2D)	(None, 256, 256, 32)	9248	activation[0][0]
batch_normalization_1	(BatchNor	(None, 256, 256, 32)	128	conv2d_1[0][0]
activation_1	(Activation)	(None, 256, 256, 32)	0	batch_normalization_1[0][0]
max_pooling2d	(MaxPooling2D)	(None, 128, 128, 32)	0	activation_1[0][0]
conv2d_2	(Conv2D)	(None, 128, 128, 64)	18496	max_pooling2d[0][0]
batch_normalization_2	(BatchNor	(None, 128, 128, 64)	256	conv2d_2[0][0]
activation_2	(Activation)	(None, 128, 128, 64)	0	batch_normalization_2[0][0]
conv2d_3	(Conv2D)	(None, 128, 128, 64)	36928	activation_2[0][0]

Layer	Type	Output Shape	Param	Connected to
batch_normalization_3	(BatchNor	(None, 128, 128, 64)	256	conv2d_3[0][0]
activation_3	(Activation)	(None, 128, 128, 64)	0	batch_normalization_3[0][0]
max_pooling2d_1	(MaxPooling2D)	(None, 64, 64, 64)	0	activation_3[0][0]
conv2d_4	(Conv2D)	(None, 64, 64, 128)	73856	max_pooling2d_1[0][0]
batch_normalization_4	(BatchNor	(None, 64, 64, 128)	512	conv2d_4[0][0]
activation_4	(Activation)	(None, 64, 64, 128)	0	batch_normalization_4[0][0]
conv2d_5	(Conv2D)	(None, 64, 64, 128)	147584	activation_4[0][0]
batch_normalization_5	(BatchNor	(None, 64, 64, 128)	512	conv2d_5[0][0]
activation_5	(Activation)	(None, 64, 64, 128)	0	batch_normalization_5[0][0]
max_pooling2d_2	(MaxPooling2D)	(None, 32, 32, 128)	0	activation_5[0][0]
conv2d_6	(Conv2D)	(None, 32, 32, 256)	295168	max_pooling2d_2[0][0]
batch_normalization_6	(BatchNor	(None, 32, 32, 256)	1024	conv2d_6[0][0]
activation_6	(Activation)	(None, 32, 32, 256)	0	batch_normalization_6[0][0]
conv2d_7	(Conv2D)	(None, 32, 32, 256)	590080	activation_6[0][0]

Layer	Type	Output Shape	Param	Connected to
batch_normalization_7	(BatchNor	(None, 32, 32, 256)	1024	conv2d_7[0][0]
activation_7	(Activation)	(None, 32, 32, 256)	0	batch_normalization_7[0][0]
max_pooling2d_3	(MaxPooling2D)	(None, 16, 16, 256)	0	activation_7[0][0]
conv2d_8	(Conv2D)	(None, 16, 16, 512)	1180160	max_pooling2d_3[0][0]
batch_normalization_8	(BatchNor	(None, 16, 16, 512)	2048	conv2d_8[0][0]
activation_8	(Activation)	(None, 16, 16, 512)	0	batch_normalization_8[0][0]
conv2d_9	(Conv2D)	(None, 16, 16, 512)	2359808	activation_8[0][0]
batch_normalization_9	(BatchNor	(None, 16, 16, 512)	2048	conv2d_9[0][0]
activation_9	(Activation)	(None, 16, 16, 512)	0	batch_normalization_9[0][0]
max_pooling2d_4	(MaxPooling2D)	(None, 8, 8, 512)	0	activation_9[0][0]
conv2d_10	(Conv2D)	(None, 8, 8, 1024)	4719616	max_pooling2d_4[0][0]
batch_normalization_10	(BatchNo	(None, 8, 8, 1024)	4096	conv2d_10[0][0]
activation_10	(Activation)	(None, 8, 8, 1024)	0	batch_normalization_10[0][0]
conv2d_11	(Conv2D)	(None, 8, 8, 1024)	9438208	activation_10[0][0]

Layer	Type	Output Shape	Param	Connected to
batch_normalization_11	(BatchNo	(None, 8, 8, 1024)	4096	conv2d_11[0][0]
activation_11	(Activation)	(None, 8, 8, 1024)	0	batch_normalization_11[0][0]
conv2d_transpose	(Conv2DTranspo	(None, 16, 16, 512)	2097664	activation_11[0][0]
Concatenate	(Concatenate)	(None, 16, 16, 1024)	0	activation_9[0][0] conv2d_transpose_1[0][0]
batch_normalization_12	(BatchNo	(None, 16, 16, 1024)	4096	concatenate[0][0]
activation_12	(Activation)	(None, 16, 16, 1024)	0	batch_normalization_12[0][0]
conv2d_12	(Conv2D)	(None, 16, 16, 512)	4719104	activation_12[0][0]
batch_normalization_13	(BatchNo	(None, 16, 16, 512)	2048	conv2d_12[0][0]
activation_13	(Activation)	(None, 16, 16, 512)	0	batch_normalization_13[0][0]
conv2d_13	(Conv2D)	(None, 16, 16, 512)	2359808	activation_13[0][0]
batch_normalization_14	(BatchNo	(None, 16, 16, 512)	2048	conv2d_13[0][0]
activation_14	(Activation)	(None, 16, 16, 512)	0	batch_normalization_14[0][0]
conv2d_transpose_1	(Conv2DTrans	(None, 32, 32, 256)	524544	activation_14[0][0]

Layer	Type	Output Shape	Param	Connected to
concatenate_1	(Concatenate)	(None, 32, 32, 512)	0	activation_7[0][0]
				conv2d_transpose_1[0][0]
batch_normalization_15	(BatchNo	(None, 32, 32, 512)	2048	concatenate_1[0][0]
activation_15	(Activation)	(None, 32, 32, 512)	0	batch_normalization_15[0][0]
conv2d_14	(Conv2D)	(None, 32, 32, 256)	1179904	activation_15[0][0]
batch_normalization_16	(BatchNo	(None, 32, 32, 256)	1024	conv2d_14[0][0]
activation_16	(Activation)	(None, 32, 32, 256)	0	batch_normalization_16[0][0]
conv2d_15	(Conv2D)	(None, 32, 32, 256)	590080	activation_16[0][0]
batch_normalization_17	(BatchNo	(None, 32, 32, 256)	1024	conv2d_15[0][0]
activation_17	(Activation)	(None, 32, 32, 256)	0	batch_normalization_17[0][0]
conv2d_transpose_2	(Conv2DTrans	(None, 64, 64, 128)	131200	activation_17[0][0]
concatenate_2	(Concatenate)	(None, 64, 64, 256)	0	activation_5[0][0] conv2d_transpose_2[0][0]
batch_normalization_18	(BatchNo	(None, 64, 64, 256)	1024	concatenate_2[0][0]

Layer	Type	Output Shape	Param	Connected to
activation_18	(Activation)	(None, 64, 64, 256)	0	batch_normalization_18[0][0]
conv2d_16	(Conv2D)	(None, 64, 64, 128)	295040	activation_18[0][0]
batch_normalization_19	(BatchNo	(None, 64, 64, 128)	512	conv2d_16[0][0]
activation_19	(Activation)	(None, 64, 64, 128)	0	batch_normalization_19[0][0]
conv2d_17	(Conv2D)	(None, 64, 64, 128)	147584	activation_19[0][0]
batch_normalization_20	(BatchNo	(None, 64, 64, 128)	512	conv2d_17[0][0]
activation_20	(Activation)	(None, 64, 64, 128)	0	batch_normalization_20[0][0]
conv2d_transpose_3	(Conv2DTrans	(None, 128, 128, 64)	32832	activation_20[0][0]
concatenate_3	(Concatenate)	(None, 128, 128, 128	0	activation_3[0][0] conv2d_transpose_3[0][0]
batch_normalization_21	(BatchNo	(None, 128, 128, 128	512	concatenate_3[0][0]
activation_21	(Activation)	(None, 128, 128, 128	0	batch_normalization_21[0][0]
conv2d_18	(Conv2D)	(None, 128, 128, 64)	73792	activation_21[0][0]
batch_normalization_22	(BatchNo	(None, 128, 128, 64)	256	conv2d_18[0][0]

Layer	Type	Output Shape	Param	Connected to
activation_22	(Activation)	(None, 128, 128, 64)	0	batch_normalization_22[0][0]
conv2d_19	(Conv2D)	(None, 128, 128, 64)	36928	activation_22[0][0]
batch_normalization_23	(BatchNo	(None, 128, 128, 64)	256	conv2d_19[0][0]
activation_23	(Activation)	(None, 128, 128, 64)	0	batch_normalization_23[0][0]
conv2d_transpose_4	(Conv2DTrans	(None, 256, 256, 32)	8224	activation_23[0][0]
concatenate_4	(Concatenate)	(None, 256, 256, 64)	0	activation_1[0][0] conv2d_transpose_4[0][0]
batch_normalization_24	(BatchNo	(None, 256, 256, 64)	256	concatenate_4[0][0]
activation_24	(Activation)	(None, 256, 256, 64)	0	batch_normalization_24[0][0]
conv2d_20	(Conv2D)	(None, 256, 256, 32)	18464	activation_24[0][0]
batch_normalization_25	(BatchNo	(None, 256, 256, 32)	128	conv2d_20[0][0]
activation_25	(Activation)	(None, 256, 256, 32)	0	batch_normalization_25[0][0]
conv2d_21	(Conv2D)	(None, 256, 256, 32)	9248	activation_25[0][0]
batch_normalization_26	(BatchNo	(None, 256, 256, 32)	128	conv2d_21[0][0]

Layer	Type	Output Shape	Param	Connected to
activation_26	(Activation)	(None, 256, 256, 32)	0	batch_normalization_26[0][0]
conv2d_22	(Conv2D)	(None, 256, 256, 1)	33	activation_26[0][0]
Total params: 31,129,377				
Trainable params: 31,113,377				
Non-trainable params: 16,000				