ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

BÁO CÁO

PHÂN LỚP LÚA SỬ DỤNG ẢNH SENTINEL-1A VÀ SENTINEL-2 TRÊN ĐỒNG BẰNG SÔNG HỒNG

Họ và tên: Ngô Minh Hoàng MSSV: 16020064

Họ và tên: Lê Hoàng MSSV: 16020229

Họ và tên: Nguyễn Văn Hoàng MSSV: 16020231

Lớp: K61-CA-CLC1

Người hướng dẫn: Phan Anh

HÀ NỘI – 2019

Mục lục

1. Giới thiệu chung

1.1. Một số khái niệm cơ bản

- Sentinel là tên của một loạt các vệ tinh quan sát trái đất thuộc Chương trình Copernicus của Cơ quan Không gian Châu Âu (ESA).

  + **Sentinel-1A** là vệ tinh dầu tiên trong loạt các vệ tinh thuộc chương trình Copernicus, có nhiệm vụ giám sát băng, tràn dầu, gió và sóng biển, thay đổi sử dụng đất, biến dạng địa hình và đáp ứng các trường hợp khẩn cấp lũ và động đất. Do là dữ liệu radar nên có các chế độ phân cực đơn VV hoặc HH) và phân cực đôi (VV+VH hoặc HH+HV).

  + **Sentinel-2** là vệ tinh không gian giám sát trái đất hỗ trợ các dịch vụ như: theo dõi rừng, sự thay đổi lớp phủ, và quản lý thiên tai. Hệ thống này bao gồm 2 vệ tinh (2A và 2B), không có chế độ phân cực như sentinel-1A.

- Chỉ số thực vật **NDVI (normalized difference vegetation index)**:

  NDVI = (NIR-RED)/(NIR+RED)

  Trong đó IR là giá trị bức xạ của bước sóng cận hồng ngoại (**near infrared**), RED là kênh đỏ tương ứng với Band 8 và Band 4 của dữ liệu Sentinel2-.Chỉ số thực vật được dùng rất rộng rãi để xác định mật độ phân bố của thảm thực vật, đánh giá trạng thái sinh trưởng và phát triển của cây trồng, làm cơ sở số liệu để dự báo sâu bệnh, hạn hán, diện tích năng suất và sản lượng cây trồng.

1.2. Mô tả bài toán

1.2.1. Nêu vấn đề

Sự phát triển một cách bùng nổ của trí tuệ nhân tạo trong những năm gần đây, đặc biệt phải kể tới học máy (machine learning) đã giúp cuộc sống của chúng ta được hưởng lợi rất nhiều khi mà những ứng dụng thực tiễn của nó len lỏi và can thiệp vào rất nhiều vấn đề trong cuộc sống, trong các lĩnh vực, ngành nghề. Machine learning cho phép những chiếc máy tính tưởng chừng như chỉ có thể chạy theo những dòng code cứng nhắc được lập trình sẵn, nay có thể tự học hỏi và đưa ra những quyết định với dữ liệu mà hệ thống nhận được. Đáng chú ý, khi mà những cỗ máy tính ngày càng được nâng tầm sức mạnh với thông số cấu hình mạnh mẽ, song song với thực tế là lượng dữ liệu chúng ta thu thập được để số hóa và cần được xử lý ngày càng nhiều, machine learning càng chứng tỏ được vai trò và tầm ảnh hưởng của mình trong việc hỗ trợ con người giải các bài toán khó. Trong số đó, việc ứng dụng các thuật toán machine learning để có thể xử lý ảnh viễn thám hiệu quả và nhanh hơn chính là một ví dụ tiêu biểu cho sức mạnh của trí tuệ nhân tạo hay cụ thể hơn là machine learning.

Trong những năm vừa qua, các cơ quan vũ trụ đã triển khai một số lượng lớn các vệ tinh lên quỹ đạo Trái Đất. Do một lượng lớn thông tin từ các vệ tinh ảnh viễn thám cung cấp, người sử dụng có điều kiện tiếp cận với nhiều loại dữ liệu ảnh vệ tinh viễn thám khác nhau, từ quang học đến radar, từ đơn phổ đến đa phổ, từ thương mại đến miễn phí. Tuy nhiên, để làm cho việc xử lý hình ảnh vệ tinh viễn thám hiệu quả hơn, việc phát triển các phương pháp phân tích hoàn toàn tự động là rất quan trọng. Do vậy, machine learning với các thuật toán khác nhau được tin tưởng sẽ cung cấp tiềm năng để giúp con người phân loại hình ảnh không gian hiệu quả hơn và nhanh hơn.

1.2.2. Vấn đề cần giải quyết

Trong giới hạn môn học Chuyên đề công nghệ này, nhóm chúng em mong muốn áp dụng thuật toán Random Forest để phân lớp lúa trên khu vực đồng bằng sông Hồng (Việt Nam) trong giai đoạn từ tháng 02/2018 đến 05/2018 dựa trên những đặc trưng thu được từ ảnh vệ tinh Sentinel 1A và Sentinel 2 (level 1C). Mục tiêu mà chúng em hướng đến bên cạnh việc áp dụng thuật toán Random Forest phân loại thành công khu vực có lúa/không có lúa, còn là so sánh độ chính xác của bộ phân lớp qua 4 cách kết hợp đặc trưng khác nhau.

1.3. Mức độ quan trọng của bài toán

Lúa là một trong những cây lương thực phổ biến nhất của thế giới, cung cấp nguồn lương thực thực phẩm cho hàng trăm triệu, thậm chí hàng tỉ người mỗi ngày. Ở Việt Nam, lúa gạo quan trọng không chỉ bởi đây là nguồn lương thực chính của người dân mà còn là sản phẩm được xuất khẩu tới hàng trăm thị trường khác nhau, thu về hàng trăm triệu USD mỗi năm. Tuy nhiên, những năm gần đây, do ảnh hưởng của quá trình đô thị hóa và ảnh hưởng của biến đổi khí hậu mà diện tích canh tác lúa gạo đang có chiều hướng giảm dần. Để đảm bảo được an ninh lương thực, nhà nước ta đã đưa ra nhiều hình thức quản lý ở cấp nhà nước đến tận các vùng canh tác lúa có thể báo cáo kịp thời hoạt động sản xuất lúa và theo dõi sản lượng hàng năm. Nhưng việc quản lý tới từng địa phương đang là một bài toán lớn cho các đơn vị các cơ quan chức năng quản lý về lương thực thực sự rất khó khăn về giám sát. Do đó, bài toán giám sát các khu vực trồng lúa áp dụng công nghệ một hướng hiệu quả là rất cần thiết đảm bảo an ninh lương thực của Việt Nam.

Việc khảo sát bằng hệ thống thông tin địa lý cung cấp cho người dùng kết quả của việc phân tích ảnh vệ tinh mang tính khách quan không bị chi phối bởi ý nghĩ chủ quan của người điều tra, những dữ liệu ảnh vệ tinh sẽ có thể đánh giá được mức đột thiệt hại của lúa do dịch hại gây ra khi đã xảy ra dịch. Việc khai tách ứng dụng của hệ thống thông tin địa lý giúp nhanh chóng bắt được tiến độ xuống giống, tiến độ thu hoạch lúa, tình hình sâu bệnh, tình hình thiên tai trong từng vụ lúa để có được kế hoạch chỉ đạo sản xuất kịp thời, khuyên cáo, định hướng cho bà con nông dân về lịch gieo cấy sử dụng giống thích hợp nhằm nâng cao hiệu quả sản xuất lúa. Nắm bắt chính

xác tiến độ sản xuất để có thể thực hiện chính sách hỗ trợ sản xuất lúa của Chính phủ đúng lúc, đúng chỗ. Theo dõi biến động của cơ cấu giống lúa qua các năm để tìm hiểu nguyên nhân, đánh giá đúng giá trị của các giống lúa, có chính sách khuyến khích cần thiết đề bà con nông dân sử dụng giống đem lại lợi ích cao nhất. Theo dõi, nắm bắt được hướng chuyển dịch công trồng của bà con nông dân để có chính sách điều chỉnh cần thiết. Trong trường hợp này, bản đồ lúa mà hệ thống thông tin địa lý cung cấp là nguồn thông tin quan trọng phục vụ việc lập chính sách điều chỉnh hướng chuyển dịch cơ cấu cây trồng của bà con nông dân. Việc sử dụng ảnh vệ tinh Sentinel 1A và Sentinel 2 có thể giúp chúng ta đưa ra những phân tích và số liệu nhanh và chính xác kể cả khi không có những dữ liệu cập nhật từ các địa phương lên.

1.4. Các công việc cần thực hiện

Qua quá trình tìm hiểu các công trình, các bài báo liên quan và nhận được sự hướng dẫn của anh Phan Anh (thành viên lab FIMO), chúng em xác định các công việc mà nhóm cần thực hiện bao gồm:

- Tiền xử lý và tải ảnh Sentinel-1A và Sentinel-2 chụp khu vực đồng bằng sông Hồng;
- Sử dụng công cụ hỗ trợ (ENVI Classic) để trích xuất ra dữ liệu từ các ảnh đã tải về;
- Tiền xử lý dữ liệu sử dụng phần mềm Microsoft Excel;
- Cài đặt thuật toán Random Forest;
- Thực hiện huấn luyện dữ liệu và chạy với dữ liệu test của bộ dữ liệu ảnh Sentinel 1A, ghi lại kết quả;
- Thực hiện huấn luyện dữ liệu và chạy với dữ liệu test của bộ dữ liệu ảnh Sentinel 2 (level 1C), ghi lại kết quả;
- So sánh kết quả thu được và tiến hành viết báo cáo;

Để đảm bảo tiến độ và hoàn thành tốt đề tài, do nhóm có 3 thành viên nên chúng em phân chia các công việc chính như sau:

- Thành viên Nguyễn Văn Hoàng: tải ảnh, trích xuất đặc trưng VH, VV của ảnh Sentinel-1A;
- Thành viên Lê Hoàng: tải ảnh, trích xuất đặc trưng NDVI của ảnh Sentinel-2;
- Thành viên Ngô Minh Hoàng: thực hiện huấn luyện và đánh giá model sử dụng thuật toán Random Forest;

2. Các nghiên cứu liên quan

Việc áp dụng các thuật toán machine learning để phân lớp lúa trên ảnh vệ tinh đã và đang thu hút rất nhiều sự quan tâm của cộng đồng nghiên cứu. Tính đến nay, năm 2019, đã có rất nhiều công trình nghiên cứu liên quan tới chủ đề này được công bố, đem lại nhiều thành tựu không chỉ cho ngành khoa học viễn thám, mà còn đóng góp phần quan trọng giúp các quốc gia xây dựng nền nông nghiệp 4.0 một cách bền vững.

Trong số các công trình đó, [1] là công trình mới nhất và tương đối quy mô khi áp dụng các thuật toán machine learning đối với dữ liệu ảnh đa thời gian lấy từ vệ tinh Sentinel 1A và Landsat để mô phỏng các cánh đồng lúa ở một khu vực của Trung Quốc.

Công trình [1] sử dụng hai thuật toán học máy là Random Forest (RF) và Support Vector Machine (SVM) trên bộ dữ liệu kết hợp giữa ảnh Sentinel-1A SAR đa thời gian (phân cực VV và VH) với ảnh NDVI của vệ tinh Landsat để phân loại vùng trồng lúa và không trồng lúa, sau đó lập bản đồ lúa cho vùng đông bắc Trung Quốc. Các kết quả phân loại cho thấy sự hiệu quả của việc sử dụng đặc trưng NDVI trong quá trình phân loại giúp tăng đáng kể độ chính xác phân loại tổng thể ở cả hai loại phân lớp SVM và RF. Cách tiếp cận tối ưu nhất thu được từ nghiên cứa này là phân lớp dùng dữ liệu SAR phân cực VH kết hợp với đặc trưng NDVI time series cùng thuật toán RF đạt được giá trị lúa và tổng giá trị kappa tương ứng là 0,92 và 0,94, UA lúa và độ chính xác phân loại tổng thể là 95% và 95.2%.

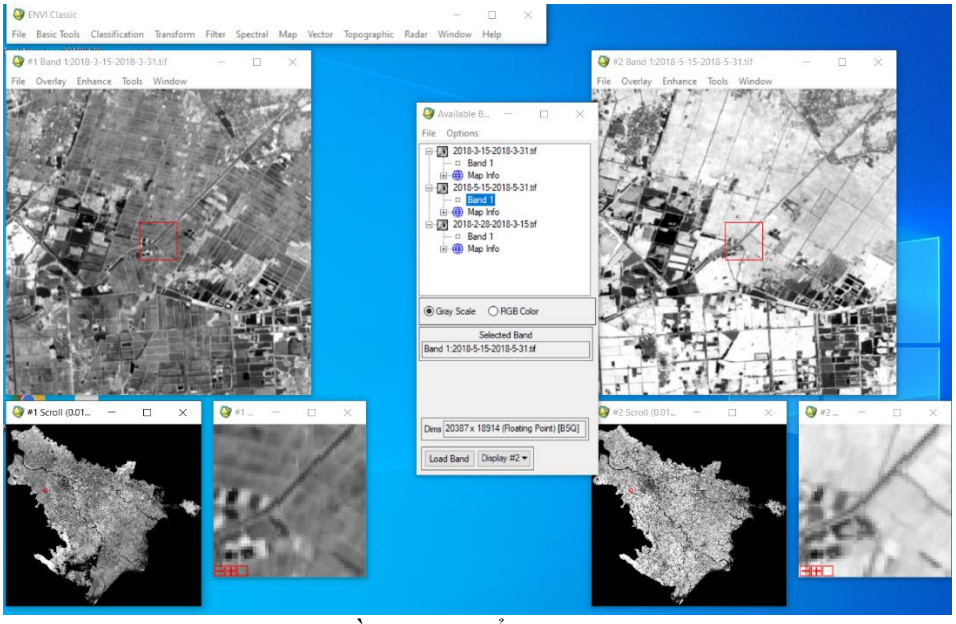
Một bước tiến từ công trình nghiên cứu [1] sẽ là triển khai phương pháp RF trên cơ sở hoạt động tạo ra các bản đồ cánh đồng lúa có độ phân giải không gian tốt dựa trên dữ liệu SAR của Sentinel-1A kết hợp với thông tin hiện tượng học dựa trên NDVI. Ngoài ra, cả hai cách phân loại trên đều được đánh giá có thể được triển khai với dữ liệu đào tạo mặt đất hạn chế, sử dụng thuật toán học máy cho cách tiếp cận hiệu quả, nhanh và mạnh so với các bộ phân loại dựa trên pixel truyền thống khác như thuật toán tối đa khả năng (maximum likelihood). Trong trường hợp không có thông tin về hiện tượng, nghiên cứu tiếp tục cho thấy sự thích hợp của việc sử dụng cả hai kênh phân cực đa thời gian (VH và VV) trong quy trình phân loại với thuật toán học máy SVM hoặc RF và tiềm năng của kho lưu trữ dữ liệu SAR đa chiều thường xuyên truy cập mở, được cung cấp bởi các vệ tinh Sentinel-1A và 1B hứa hẹn là cơ sở để lập bản đồ vùng trồng lúa từ quy mô khu vực đến toàn cầu.

3. Phương pháp

3.1. Phương pháp sử dụng

3.1.1. Dữ liệu đa thời gian

Lúa là cây trồng có sự biến động và thay đổi cao. Đặc điểm quang phổ của lúa gạo thay đổi khá lớn trong suốt vòng sinh trưởng của lúa từ lúa nước đến chín vàng và thu hoạch. Để việc phân lớp lúa được chính xác thì đòi hỏi phải thường xuyên quan sát và thu thập ảnh chụp cánh đồng lúa. Do đó, trong đề tài này, nhóm chúng em đã sử dụng hình ảnh ghép hàng tháng của cả hai loại ảnh Sentinel-1A và Sentinel-2 để có thể phân loại lúa. Cụ thể chúng em đã lấy ảnh Sentinel-1A và Sentinel-2 đa thời gian, kéo dài từ tháng 02/2018 tới hết tháng 05/2018, tức là một vụ mùa lúa.



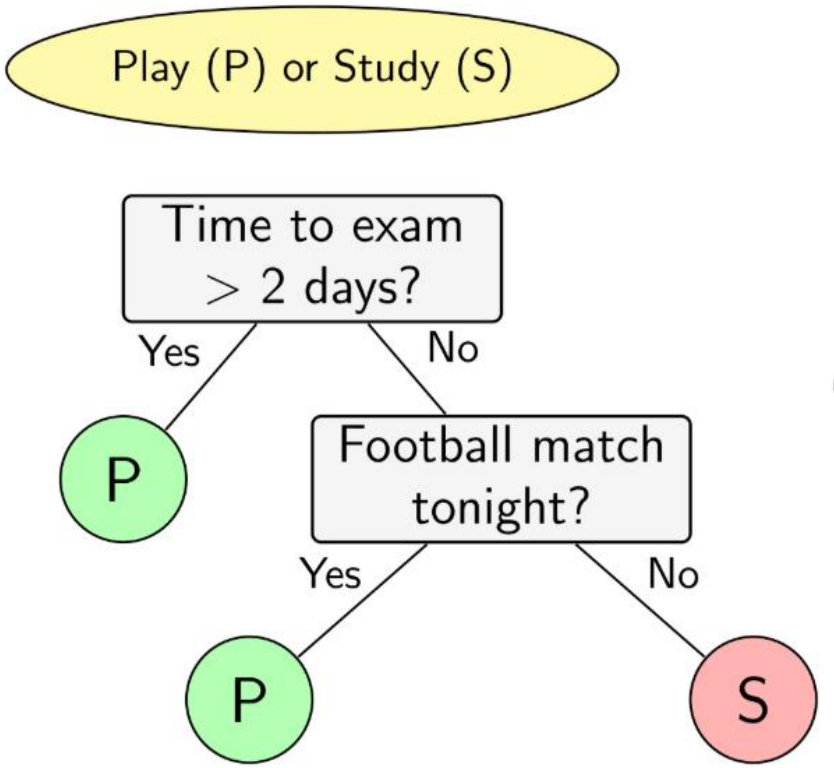
Hình 1. Ví dụ về sự thay đổi của lúa theo thời gian

3.1.2. Bộ phân lớp

Nhóm chúng em xác định bài toán cần giải quyết thuộc nhóm bài toán **phân loại (classification)**. Do đó, thay vì sử dụng thuật toán nổi tiếng và được sử dụng rộng rãi là **k-nearest neighbors,** chúng em sử dụng thuật toán **Random Forest** với mong muốn có được độ chính xác tốt hơn. Đây là một thuật toán còn tương đối mới, và được coi là một cuộc cách mạng trong Machine Learning. **Random Forest** chỉ phức tạp hơn một chút so với **k-nearest neighbors**, nhưng nó hiệu quả hơn nhiều nếu xét trên hiệu năng tính toán của máy tính. Bên cạnh đó, **Random Forest** còn cho kết quả chính xác hơn nhiều so với **k-nearest neighbors**. Nghĩa là, khi các nhà nghiên cứu kiểm thử kết quả của hai thuật toán này trên các bộ dữ liệu khác nhau, **Random Forest** thường cho kết quả đúng hơn so với **k-nearest neighbors**.

3.1.2.1. Thuật toán Random Forest

3.1.2.1.1. Decision Tree



Hình 2. Ví dụ về cây quyết định

Việc quan sát, suy nghĩ và ra các quyết định của con người thường được bắt đầu từ các câu hỏi. Machine learning cũng có một mô hình ra quyết định dựa trên các câu hỏi. Mô hình này có tên là cây quyết định (***decision tree***).

Trong *decision tree*, các ô màu xám, lục, đỏ trên Hình 1 được gọi là các node. Các node thể hiện đầu ra (màu lục và đỏ) được gọi là node lá (*leaf node* hoặc *terminal node*). Các node thể hiện câu hỏi là các *non-left node*. *Non-leaf node* trên cùng (câu hỏi đầu tiên) được gọi là node gốc (*root node*). Các *non-leaf node* thường có hai hoặc nhiều con (*child node*). Các child node này có thể là một *leaf node* hoặc một *non-leaf node* khác. Các *child node* có cùng bố mẹ được gọi là sibling node. Nếu tất cả các *non-leaf node* chỉ có hai *child node*, ta nói rằng đó là một *binary decision tree* (cây quyết định nhị phân). Các câu hỏi trong *binary decision tree* đều có thể đưa được về dạng câu hỏi đúng hay sai. Các *decision tree* mà một *leaf node* có nhiều *child node* cũng có thể được đưa về dạng một *binary decision tree*. Điều này có thể đạt được vì hầu hết các câu hỏi đều có thể được đưa về dạng câu hỏi đúng sai.

Ở bài toán chúng em đặt ra, mỗi một *non-leaf node* thể hiện cho một truy vấn về một thuộc tính của lúa trên ảnh vệ tinh và một *leaf node* thể hiện kết luận là lúa hay không phải lúa.

3.1.2.1.2. Random Forest

Random Forest (rừng ngẫu nhiên) là một thuật toán học có giám sát (**supervised learning**) và là một thành viên trong họ thuật toán **decision tree** (cây quyết định). Ý tưởng của thuật toán *Random Forest* như sau: Thuật toán này sinh một số cây quyết định (thường là vài trăm) và sử dụng chúng. Các câu hỏi của cây quyết định sẽ là câu hỏi về các thuộc tính. Ví dụ: *"Số liệu VV+VH của vị trí trên ảnh có lớn hơn -1 hay không?"*. Câu giá trị ở nút lá sẽ là các phân lớp (là lúa hay không phải lúa). Sử dụng hàng trăm cây quyết định là bất khả thi với con người, nhưng máy tính có thể làm việc này tương đối dễ dàng.

Để tạo ra một cây quyết định, thuật toán *Random Forest* luôn bắt đầu bằng một cây rỗng. Một cây quyết định rỗng chỉ có một ô *Start* chỉ thẳng đến câu trả lời (ô xanh lá). Tiếp theo, thuật toán sẽ tìm câu hỏi đầu tiên và bắt đầu xây dựng cây quyết định. Mỗi lần thuật toán tìm được thêm một câu hỏi, nó tạo hai nhánh trên cây quyết định. Khi không còn câu hỏi nào nữa, thuật toán dừng lại và chúng ta có một cây quyết định hoàn chỉnh.

Làm thế nào để tìm ra những câu hỏi tốt nhất cho cây quyết định? Đây là một bước khá phức tạp nhưng ý tưởng đằng sau nó tương đối đơn giản: Ở thời điểm bắt đầu, thuật toán của chúng ta chưa biết phân biệt một vị trí trên ảnh là lúa hay không phải lúa. Để tìm ra câu hỏi tốt nhất, thuật toán thử đưa ra tất cả các câu hỏi có thể (có khi là hàng triệu câu hỏi). Ví dụ: "Vị trí đấy có số liệu VH+VV ngày đầu tiên trong tháng 2 là bao nhiêu?",... Sau đó, với mỗi câu hỏi, thuật toán sẽ đánh giá mức độ hiệu quả mà câu hỏi này giúp phân biệt các chủng loại, hay các *class*. Câu hỏi được chọn không cần thiết phải hoàn hảo, nhưng nó phải tốt hơn những câu hỏi khác. Để tính toán mức độ hiệu quả của câu hỏi, chúng ta sử dụng một độ đo có tên là **information gain**. Câu hỏi với *information gain* lớn nhất sẽ được chọn như là câu hỏi tốt nhất để xây dựng cây quyết định.

Chúng ta sẽ xây dựng nhiều cây quyết định một cách ngẫu nhiên dựa vào 2 quá trình sau đây:

- Để chắc chắn rằng không phải tất cả các cây quyết định cho cùng câu trả lời, thuật toán Random Forest chọn ngẫu nhiên các đặc tính của lúa. Chính xác hơn, Random Forest sẽ xoá một số đặc tính và lặp lại một số khác một cách ngẫu nhiên. Xét toàn cục, những quan sát này vẫn rất gần với tập các quan sát ban đầu, nhưng những thay đổi nhỏ sẽ đảm bảo rằng mỗi cây quyết định sẽ có một chút khác biệt. Quá trình này gọi là **bootstrapping.**
- Thêm vào đó, để thực sự chắc chắn các cây quyết định là khác nhau, thuật toán Random Forest sẽ ngẫu nhiên bỏ qua một số câu hỏi khi xây dựng cây quyết định. Trong trường hợp này, nếu câu hỏi tốt nhất

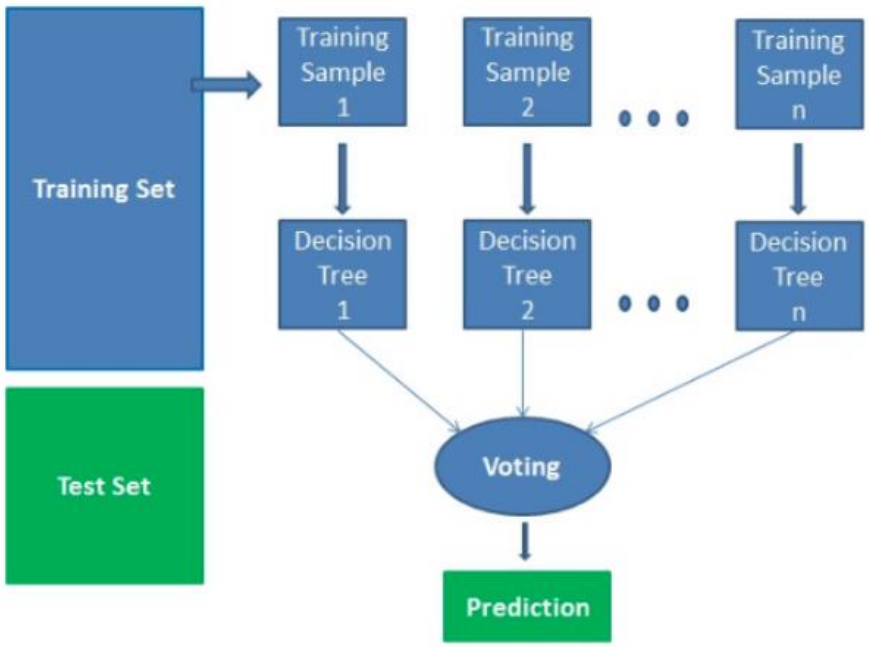
không được chọn, một câu hỏi kế tiếp sẽ được lựa chọn để dựng cây. Quá trình này được gọi là **attribute sampling**.

3.1.2.2. Điểm mạnh của thuật toán Random Forest

- Thuật toán Random Forest có thể sử dụng cho cả bài toán phân loại (classification) và hồi quy (regression);
- Random Forest làm việc được với dữ liệu thiếu giá trị;
- Khi Forest có nhiều cây hơn, chúng ta có thể tránh được việc overfitting với tập dữ liệu;
- Có thể tạo mô hình cho các giá trị phân loại;

3.1.2.3. Hoạt động của Random Forest

Random Forest hoạt động bằng cách đánh giá nhiều cây quyết định ngẫu nhiên, và lấy ra kết quả được đánh giá tốt nhất trong số kết quả trả về.



Hình 3. Hoạt động của thuật toán Random Forest

Cụ thể:

(1) Chọn các mẫu ngẫu nhiên từ tập dữ liệu đã cho;
(2) Thiết lập cây quyết định cho từng mẫu và nhận kết quả dự đoán từ mỗi quyết định cây;
(3) Bỏ phiếu cho mỗi kết quả dự đoán;
(4) Chọn kết quả được dự đoán nhiều nhất là dự đoán cuối cùng.

3.2. Cách đánh giá

Chúng em sẽ kết hợp các đặc trưng VH,VV và NDVI theo 4 trường hợp khác nhau để làm dữ liệu huấn luyện model và so sánh độ chính xác của model trong 4 trường hợp đó. Từ kết quả so sánh, chúng em sẽ xác định cách kết hợp đặc trưng tốt nhất giúp phân biệt vùng có lúa hay không có lúa.

- Trường hợp 1: Dữ liệu huấn luyện gồm 10 đặc trưng VH và 10 đặc trưng VV.
- Trường hợp 2: Dữ liệu huấn luyện gồm 10 đặc trưng VH+VV.
- Trường hợp 3: Dữ liệu huấn luyện gồm 10 đặc trưng VH, 10 đặc trưng VV và 6 đặc trưng NDVI.
- Trường hợp 4: Dữ liệu huấn luyện gồm 10 đặc trưng VH+VV và 6 đặc trưng NDVI.

4. Thực nghiệm

4.1. Thiết kế thực nghiệm

4.1.1. Tải ảnh Sentinel 1A và Sentinel 2 (L1C):

Chúng ta tải ảnh và tiền xử lí ảnh Sentinel 1A và ảnh Sentinel 2 (level 1C) bằng Google Earth Engine:

```
Imports (1 entry)
var shape: Table users/hoangle12298/shape
var mosaic_image = function (col, channel, folder, resolution, shape) {
  /**
  Input:
    col: Image collection
    channel: VV/VH
    folder: drive folder
    resolution: S1A - 10m, LS8 - 30m
    shape: boundary shapefile cua DBSH
  **/
  var maxPixels = 10000000000000;
  var colList = col.toList(500);
  var col_length = colList.size().getInfo();

  var mosaic = ee.Image();
  var img1 = ee.Image();
  var img2 = ee.Image();

  for (var i = 0; i < col_length - 1 ; i = i+2) {// Anh Sentinel 1A can 2 anh de cover het DBSH

    img1 = ee.Image(colList.get(i));
    img2 = ee.Image(colList.get(i+1));

    var id = img1.id().getInfo();
    mosaic = ee.ImageCollection.fromImages([img1, img2]).mosaic().select(channel);

    console.log(mosaic);
    Map.addLayer(mosaic,{bands: [channel]}, id);
    export2drive(mosaic.clip(shape), folder, resolution, id, shape, maxPixels);

  }
}

var export2drive = function (img, folder, scale, id, shape, maxPixels){
  Export.image.toDrive({
        image:img,
        description: id,
        folder: folder,
        fileNamePrefix: id,
        region: shape.geometry().bounds(),
        scale: scale,
        crs: 'EPSG:4326',
        maxPixels: maxPixels})
}

//main
var img = ee.ImageCollection('COPERNICUS/S1_GRD');
var sentinel = img.filterDate('2018-2-1', '2018-5-30')// Download anh tu 1/2/2018 den 30/5/2018
                  .filterBounds(shape) // Loc anh ve khu vuc ĐBSH
                  .filter(ee.Filter.eq('relativeOrbitNumber_start', 91));

var sentinel_clip = sentinel.map(function(image) {return image.clip(shape);});// Cat anh ve khu vực ĐBSH
console.log(sentinel_clip);

var vv = 'VV';
var vh = 'VV';
var folder = "Sentinel-1A"; // Tao folder Sentinel-1A tren drive
//var shape = ee.FeatureCollection("users/hoangle12298/DBSH_Shape"); // Doan nay import shapefile cua DBSH

mosaic_image(sentinel_clip, vv, folder, 10, shape); // Xong VV thi chay cho VH cho de kiem soat anh do id anh trung nhau
Map.addLayer(shape, {color: 'FF0000'}, 'DBSH');
```

Hình 4. Đoạn code tải ảnh Sentinel-1A

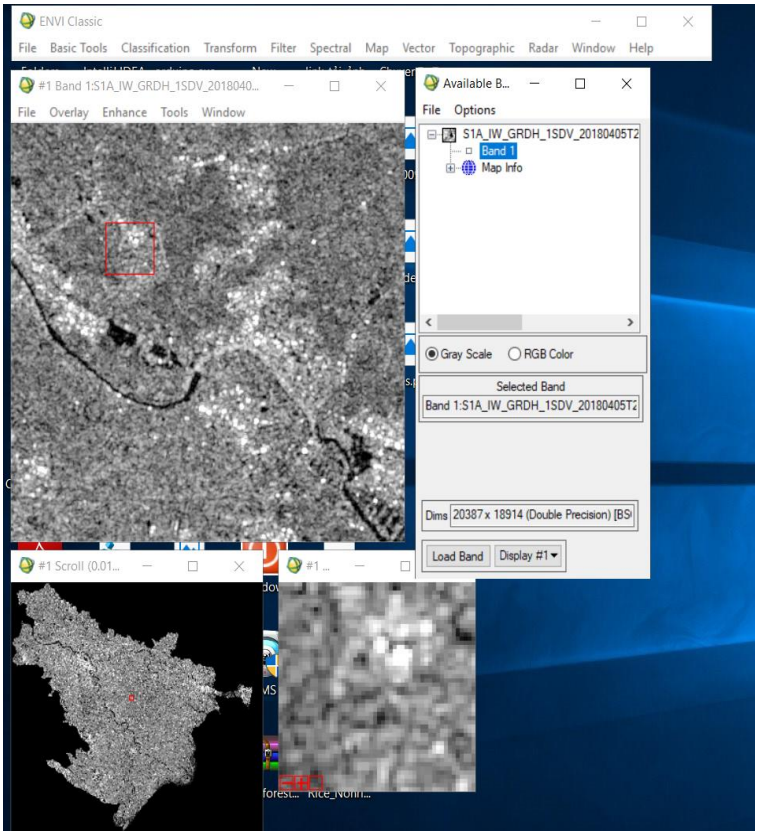
```
var export2drive = function (img, folder, scale, id, shape, maxPixels){
  Export.image.toDrive({
        image:img,
        description: img.id().getInfo(),
        folder: folder,
        fileNamePrefix: id + '_' + img.id().getInfo(),
        region: shape.geometry().bounds(),
        scale: scale,
        crs: 'EPSG:4326',
        maxPixels: maxPixels})
}
function maskS2clouds(image) {
  var qa = image.select('QA60');

  // Bits 10 and 11 are clouds and cirrus, respectively.
  var cloudBitMask = 1 << 10;
  var cirrusBitMask = 1 << 11;

  // Both flags should be set to zero, indicating clear conditions.
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0)
      .and(qa.bitwiseAnd(cirrusBitMask).eq(0));

  return image.updateMask(mask).divide(10000);
}

function get_dataset(date1, date2, shape) {
  var dataset = ee.ImageCollection('COPERNICUS/S2')
                    .filterDate(date1, date2)
                    .filterBounds(shape)
                    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 90))
                    .map(maskS2clouds);
  var dataset_clip = dataset.map(function (image) {return image.clip(shape)})
  var ndvi = dataset_clip.map(function (image) {
    var nir = image.select('B8');
    var red = image.select('B4')
    return nir.subtract(red).divide(nir.add(red)).rename('NDVI');
  });

  return ndvi.mean();
}

var dbsh = ee.FeatureCollection('users/hoangle12298/shape');

var list_date = ['2018-2-1',
                    '2018-2-15',
                    '2018-2-28',
                    '2018-3-15',
                    '2018-3-31',
                    '2018-4-15',
                    '2018-4-30',
                    '2018-5-15',
                    '2018-5-31'];

var ndviParams = {min: -1, max: 1, palette: ['blue', 'white', 'green']};

Map.addLayer(dbsh);

//mosaicking and downloading
for (var i = 0; i < list_date.length - 1 ; i++) {
  var img = get_dataset(list_date[i], list_date[i+1], dbsh);
  var id = list_date[i] + "-" + list_date[i+1]
  Map.addLayer(img.select('NDVI'), ndviParams, id);
  export2drive(img.select('NDVI'), "Sentinel2", 10, id, dbsh, 10000000000000)
}
```

Hình 5. Đoạn code tải ảnh Sentinel-2

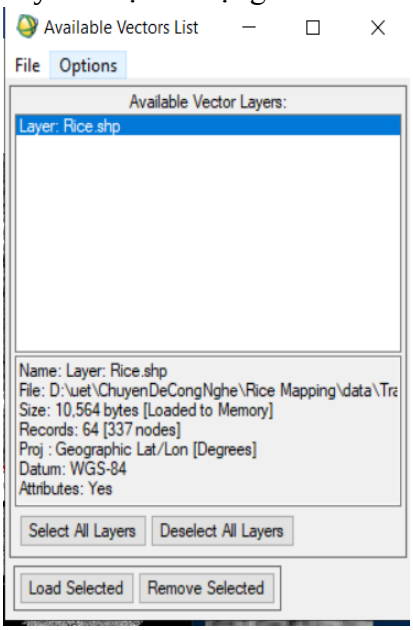
4.1.2. Trích xuất đặc trưng từ ảnh

Chúng ta sẽ trích xuất đặc trưng VH, VV của ảnh Sentinel 1A và NDVI của ảnh Sentinel 2 (level 1C) bằng phần mềm ENVI Classic như sau:

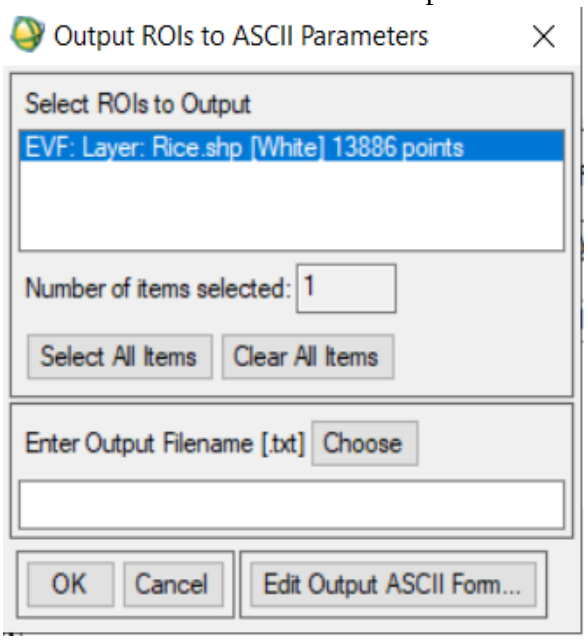
Bước 1: Load ảnh và các band ảnh



Bước 2: Load những điểm được đánh dấu là lúa (hoặc không phải lúa) vào ROI Tool để lấy số liệu về phổ VV, VH hoặc NDVI tại những điểm này. Những dữ liệu này sẽ được sử dụng để train và test model



Bước 3: Từ ROI Tool ta xuất output ra file excel



Bước 4: Ta thu được dữ liệu là đặc trưng VV, VH hoặc NDVI của các điểm sử dụng cho train và test model sau này.

	A	B	C	D	E	F	G	H	I	J
1	ID	X	Y			NDVI				Label
2	1	2623	1373	0.1842	0.4028	0.3539	0.5842	9999	0.5352	1
3	2	2619	1373	0.2208	0.5770	0.4248	0.5074	9999	0.5738	1
4	3	2620	1373	0.2095	0.6025	0.4433	0.5198	9999	0.5918	1
5	4	2621	1373	0.1981	0.5704	0.4447	0.5426	9999	0.5887	1
6	5	2622	1373	0.1803	0.5113	0.4090	0.5957	9999	0.5804	1
7	6	2624	1373	0.1825	0.4166	0.3790	0.5672	9999	0.4913	1
8	7	2623	1374	0.1695	0.4409	0.3608	0.5926	9999	0.5481	1
9	8	2621	1374	0.1906	0.5459	0.4307	0.5261	9999	0.5853	1
10	9	2619	1374	0.2086	0.5205	0.4105	0.5288	9999	0.5279	1
11	10	2620	1374	0.2035	0.5485	0.4345	0.4956	9999	0.5770	1
12	11	2622	1374	0.1696	0.5399	0.4168	0.5914	9999	0.5803	1
13	12	2624	1374	0.1705	0.3949	0.3502	0.5671	9999	0.5194	1
14	13	2622	1375	0.1754	0.4912	0.4116	0.5778	9999	0.5723	1
15	14	2623	1375	0.1772	0.4742	0.3979	0.5804	9999	0.5525	1
16	15	2619	1375	0.1966	0.4733	0.3784	0.5309	9999	0.5216	1
17	16	2620	1375	0.2016	0.4258	0.3988	0.5575	9999	0.5536	1
18	17	2621	1375	0.1891	0.4200	0.3863	0.5622	9999	0.5611	1
19	18	2624	1375	0.1730	0.4185	0.3386	0.5752	9999	0.5186	1
20	19	2622	1376	0.1666	0.4128	0.3624	0.5347	9999	0.5516	1
21	20	2623	1376	0.1709	0.4010	0.3713	0.5395	9999	0.5413	1
22	21	2619	1376	0.1878	0.4558	0.3697	0.5711	9999	0.5207	1
23	22	2620	1376	0.1925	0.4120	0.3584	0.5726	9999	0.5300	1
24	23	2621	1376	0.1838	0.4082	0.3383	0.5647	9999	0.5320	1
25	24	2624	1376	0.1790	0.4049	0.3533	0.5858	9999	0.5199	1
26	25	2620	1377	0.1670	0.4162	0.3576	0.5585	9999	0.5103	1
27	26	2621	1377	0.1765	0.4296	0.3527	0.5655	9999	0.5263	1
28	27	2622	1377	0.1753	0.3699	0.2883	0.5197	9999	0.5401	1
29	28	2623	1377	0.1738	0.3605	0.2982	0.5018	9999	0.5427	1

4.1.3. Kết hợp các đặc trưng làm dữ liệu huấn luyện model

Từ những đặc trưng riêng lẻ thu được ở trên, ta kết hợp các đặc trưng theo 4 trường hợp nêu ở mục 3.2 và tiến hành huấn luyện và đánh giá model.

4.1.4. Huấn luyện và đánh giá model sử dụng thuật toán Random Forest

Bước 1: Ta thiết lập dải hyperparameters cho Random Forest classifier

```
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
random_grid
```

```
{'bootstrap': [True, False],
 'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None],
 'max_features': ['auto', 'sqrt'],
 'min_samples_leaf': [1, 2, 4],
 'min_samples_split': [2, 5, 10],
 'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]}
```

Bước 2: Huấn luyện model theo dải parameters đã thiết lập

```
[ ] # Create a random forest Classifier. By convention, clf means 'Classifier'
    clf = RandomForestClassifier()

    clf_random = RandomizedSearchCV(estimator = clf, param_distributions = random_grid, n_iter = 100, cv = 3, verbose=2, random_state=42, n_jobs = -1)

    # Train the Classifier to take the training features and learn how they relate
    # to the training y (the species)
    clf_random.fit(x_train[features], y_train)
```

Bước 3: Đánh giá model

```
[ ] preds = clf_random.predict(x_test)
    pd.crosstab(y_test, preds, rownames=['Actual Species'], colnames=['Predicted Species'])
```

Actual Species \ Predicted Species	0	1
0	798	51
1	93	444

```
[ ] metrics.accuracy_score(y_test,preds)
```

```
0.8961038961038961
```

4.2. Kết quả

Trường hợp 1: Dữ liệu huấn luyện gồm 10 đặc trưng VH và 10 đặc trưng VV.

- Độ chính xác tốt nhất khi sử dụng phương pháp Cross Validation trên tập train là 92.8%.
- Độ chính xác thực tế trên tập test là 89.6%

Trường hợp 2: Dữ liệu huấn luyện gồm 10 đặc trưng VH+VV.

- Độ chính xác tốt nhất khi sử dụng phương pháp Cross Validation trên tập train 91.3%.
- Độ chính xác thực tế trên tập test là 87.0%

Trường hợp 3: Dữ liệu huấn luyện gồm 10 đặc trưng VH, 10 đặc trưng VV và 6 đặc trưng NDVI.

- Độ chính xác tốt nhất khi sử dụng phương pháp Cross Validation trên tập train là 94.2%
- Độ chính xác thực tế trên tập test là 90.5%

Trường hợp 4: Dữ liệu huấn luyện gồm 10 đặc trưng VH+VV và 6 đặc trưng NDVL.

- Độ chính xác tốt nhất khi sử dụng phương pháp Cross Validation trên tập train 92.8%.
- Độ chính xác thực tế trên tập test là 89.9%

4.3. Đánh giá

Như vậy, sau khi tiến hành thực nghiệm với 4 trường hợp nêu trên, chúng em có thể rút ra đánh giá rằng model dùng dữ liệu huấn luyện với 10 đặc trưng VH, 10 đặc trưng VV và 6 đặc trưng NDVI cho ra độ chính xác lớn nhất.

5. Kết luận

Trong phạm vi đề tài này, chúng em đã đề xuất phương pháp và cài đặt thành công mô hình phân lớp lúa cho khu vực đồng bằng sông Hồng (Việt Nam) sử dụng thuật toán Random Forest đối với dữ liệu đặc trưng VH, VV (ảnh Sentinel-1A) và NDVI (ảnh Sentinel-2). Sau đây là một số kết luận chúng em rút ra được sau quá trình thực hiện đề tài:

- Việc kết hợp dùng dữ liệu ảnh Sentinel-1A (đặc trưng VH, VV) và Sentinel-2 (đặc trưng NDVI) có tiềm năng cho việc phân lớp lúa tại đồng bằng sông Hồng;
- Bộ phân lớp Random Forest là một lựa chọn tốt cho việc phân lớp lúa. Độ chính xác lớn nhất của đạt được của bộ phân lớp là 90.5% khi dùng dữ liệu huấn luyện gồm 10 đặc trưng VH, 10 đặc trưng VV và 6 đặc trưng NDVI;

Bên cạnh đó, để việc phân lớp lúa được hiệu quả và đạt độ chính xác cao hơn, chúng em nhận ra còn nhiều vấn đề cần giải quyết:

- Lúa còn bị phân loại sai, bị nhầm với một số loài thực vật khác như cây cối, rau quả,…;
- Ảnh vệ tinh có một số ảnh chất lượng chưa đủ tốt, ảnh hưởng tới quá trình thu thập dữ liệu đặc trưng;

6. Tài liệu tham khảo

[1] Alex O. Onojeghuo, George A. Blackburn, Qunming Wang, Peter M. Atkinson, Daniel Kindred & Yuxin Miao (2018), ***Mapping paddy rice fields by applying machine learning algorithms to multi-temporal Sentinel-1A and Landsat data***, International Journal of Remote Sensing, 39:4, 1042-1067, DOI: 10.1080/01431161.2017.1395969.

[2] Nguyễn Hoàng Anh (2017), ***Nghiên cứu và phát triển phương pháp phân lớp lúa ở đồng bằng sông Hồng sử dụng ảnh vệ tinh Landsat 8***, Luận văn Thạc sĩ Công nghệ thông tin, Trường Đại học Công nghệ - Đại học Quốc gia Hà Nội.