# **RESEARCH PROPOSAL**

## **〈 Parallelization Xgboost Algorithms〉**

| **Group name:**  666  **List of members:** 19127132 – Trần Quang Duy | 19127328 – Đỗ Quốc Anh | 19127332 – Nguyễn Nam Anh |
| --- |

| **Keywords:** Accelerating the XGBoost algorithm, GPU-accelerated XGBoost , Boosting, Gradient Boosting, Plant Disease Detection, distributed computing, parallelization, ensemble algorithms.  **List of references:** Một số nguồn tham khảo của nhóm   * Documentations : [Accelerating the XGBoost algorithm using GPU computing [PeerJ]](https://peerj.com/articles/cs-127/), [GPU Accelerated XGBoost | R-bloggers](https://www.r-bloggers.com/2016/12/gpu-accelerated-xgboost/), tutorials [A Gentle Introduction to XGBoost for Applied Machine Learning - MachineLearningMastery.com](https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/), [Implementing XGBoost from scratch | by Siddhesh Jadhav | Towards Data Science](https://towardsdatascience.com/implementing-xgboost-from-scratch-6b7f2eb593c), technical reports [Out-of-Core GPU Gradient Boosting (arxiv.org)](https://arxiv.org/pdf/2005.09148.pdf), [[1603.02754] XGBoost: A Scalable Tree Boosting System (arxiv.org)](https://arxiv.org/abs/1603.02754), etc. * Source codes [A numpy/pandas implementation of XGBoost (github.com)](https://gist.github.com/Sandy4321/cc4fce780e1254fbeba32d55e0d89a58), pretrained models, demonstrative applications [Gradient Boosting (nicolas-hug.com)](http://nicolas-hug.com/pres_scipy_2019/#1), etc. * List of books, papers, or web pages [Distributed XGBoost (nvidia.com)](https://resources.nvidia.com/en-us-xgboost): |
| --- |

| **Content**   1. **Summary:**   Đồ án này nhóm làm về dự đoán bệnh thông qua hình ảnh các loại nấm sau khi xét nghiệm bằng xgboost giống như các bài toán thông thường nhưng mục đích chính là sử dụng song song hóa để cải thiện hiệu suất cho bài toán.  Kế hoạch của nhóm cho việc song song hóa sẽ là viết lại thuật toán xgboost mà không gọi thư viện của nó sau đó sẽ song song hóa những đoạn mã có thể tối ưu được bằng việc song song  Hệ thống sử dụng Google Colab, nơi cung cấp môi trường máy tính xách tay Jupyter dựa trên đám mây có quyền truy cập vào GPU bao gồm:   * Song song đa lõi: Google Colab cung cấp quyền truy cập vào các máy có nhiều lõi CPU với các thư viện Python như multiprocessing hoặc joblib để phân phối các tác vụ tính toán chuyên sâu trên nhiều lõi. Điều này có thể hữu ích để song song hóa các giai đoạn nhất định trong ứng dụng, chẳng hạn như tiền xử lý dữ liệu hoặc điều chỉnh siêu tham số. * Tính song song của GPU: Google Colab cung cấp quyền truy cập vào GPU, cụ thể là GPU NVIDIA Tesla. GPU vượt trội ở khả năng xử lý song song và có thể tăng tốc đáng kể các tác vụ học sâu, bao gồm trích xuất tính năng và đào tạo các mô hình máy học. Có thể sử dụng các khung học sâu phổ biến như TensorFlow hoặc PyTorch, được tích hợp hỗ trợ GPU, để tận dụng sức mạnh tính toán song song của GPU. Điều này có thể tăng tốc đáng kể các bước trích xuất và đào tạo đặc trưng cho đồ án..  1. **Background**   Mục tiêu của dự án là dự đoán các bệnh từ hình ảnh các loại nấm bằng cách sử dụng thuật toán XGBoost và tận dụng thuật toán song song để tăng tốc quá trình. Ứng dụng này liên quan đến việc đào tạo một mô hình học máy để phân loại nấm là khỏe mạnh hay bị bệnh dựa trên hình ảnh của chúng. Quá trình này có thể được chia thành các bước sau:   * Thu thập và tiền xử lý dữ liệu: Thu thập một bộ dữ liệu lớn về hình ảnh nấm, bao gồm cả mẫu khỏe mạnh và mẫu bị bệnh. Tiền xử lý hình ảnh bằng cách thay đổi kích thước, chuẩn hóa và tăng cường dữ liệu để cải thiện hiệu suất của mô hình. * Trích xuất đặc trưng:: Trích xuất các đặc trưng có liên quan từ các hình ảnh được xử lý trước để đại diện cho từng mẫu nấm. Bước này có thể liên quan đến các kỹ thuật như trích xuất đặc trưng dựa trên học sâu hoặc kỹ thuật tính năng thủ công. * Huấn luyện mô hình XGBoost: Sử dụng các đặc trưng được trích xuất và các nhãn tương ứng để huấn luyện mô hình XGBoost. XGBoost là một thuật toán tăng cường độ dốc phổ biến có thể xử lý dữ liệu dạng bảng với cả các đặc trưng numeric và category một cách hiệu quả. * Điều chỉnh siêu tham số: Tối ưu hóa mô hình XGBoost bằng cách điều chỉnh các siêu tham số của nó bằng các kỹ thuật như tìm kiếm dạng lưới hoặc tối ưu hóa Bayes. Bước này nhằm mục đích tìm ra tập hợp siêu đường kính tốt nhất mang lại độ chính xác dự đoán cao nhất.   Để tăng tốc ứng dụng, tính toán song song có thể được tận dụng ở các khía cạnh sau:   * Tiền xử lý dữ liệu: Vì tập dữ liệu có thể lớn và các bước tiền xử lý cần nhiều tính toán, nên có thể sử dụng phương pháp xử lý song song để phân phối khối lượng công việc trên nhiều lõi hoặc máy. Mỗi lõi hoặc máy có thể xử lý một tập hợp con của tập dữ liệu, xử lý trước hình ảnh và hợp nhất các kết quả sau đó. * Trích xuất đặc trưng: Trích xuất tính năng dựa trên học sâu, thường liên quan đến mạng thần kinh tích chập (CNN), có thể hưởng lợi rất nhiều từ điện toán song song. CNN rất tốn kém về mặt tính toán và việc tính toán song song trên nhiều GPU hoặc máy có thể tăng tốc đáng kể quá trình trích xuất tính năng. * Đào tạo mô hình XGBoost: XGBoost vốn đã hỗ trợ tính toán song song bằng cách sử dụng nhiều lõi CPU. Thuật toán có thể phân phối khối lượng công việc trên các lõi, đẩy nhanh quá trình đào tạo. Ngoài ra, các kỹ thuật đào tạo phân tán, chẳng hạn như song song dữ liệu hoặc song song mô hình, có thể được áp dụng để tận dụng nhiều máy và GPU để đào tạo nhanh hơn.  1. **The challenge**  * Về input:   Ở đồ án này input sẽ là hình ảnh và output là loại bệnh. Riêng việc xử lý với hình ảnh đã là một khó khăn lớn. Nếu không xử lý trước, loại input này làm cho số features trở nên rất lớn (độ phân giải ảnh x số kênh màu)   * Về output:   Đây là toán phân lớp, nhưng không phải phân lớp nhị phân thông thường. Dạng label này thì các thuật toán decision tree thông thường sẽ dễ dàng vận hành nhưng khi kết hợp với gradient boosting thì việc tính toán residual sẽ khó khăn hơn so với các bài toán hồi quy     * Về thuật toán:   Nếu không sử dụng thư viện có sẵn thì đây là thuật toán khá khó để xây dựng lại. Đòi hỏi phải đọc nhiều tài liệu về gradient boosting cũng như là xgb   * Về song song hóa:   Vì trong thuật toán gradient boosting ta sẽ phải tạo ra nhiều cây nhưng cây tạo sau được tạo dựa trên kết quả của cây trước đó:    Vì vậy nên việc song song ở bước này gần như là không thực hiện được. Nhưng ta có thể tiến hành song song hóa ở các bước khác nhỏ hơn.   1. Resources   a. Về dữ liệu:   * Dữ liệu được tải về từ kaggle tại [đây](https://www.kaggle.com/datasets/saroz014/plant-disease) * Sau khi giải nén ta được 38 folder con tượng trưng cho 38 loại bệnh đã được phân loại sẵn. Mỗi folder chứa hình ảnh lá của các cây bị bệnh. Các file hình có kích thước khác nhau   b. Về cài đặt mô hình:   * Phiên bản tuần tự đầu tiên được làm trên môi trường google colab bằng các thư viện numpy và pandas. Những bản song song hay tuần tự sau này cũng sẽ được xây dựng trên môi trường colab hoặc kaggle. Nhóm bắt đầu từ việc cài đặt thuật toán gradient boosting trước. Ở đây nhóm cài bằng numpy và pandas nhưng ở phiên bản đầu tiên thì phần decision tree vẫn từ sklearn cung cấp, việc cài đặt này hoàn toàn dựa vào các tài liệu được liệt kê bên dưới. Tuy nhiên ở phiên bản đầu tiên thì phần decision tree vẫn từ sklearn cung cấp * Nhóm có tham khảo tài liệu về gradient boosting tại [đây](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbkdYcmZ0LVlnTWZwa3lHTFltbWpmZzN3cTBIUXxBQ3Jtc0ttZ1JVZ3lLRmdwUERkTGlsN05oZDdxNzJ0LWlYaWtjTi1qY1k2LVFWOFVqeGtnUUtqQV9ZcTQ3S1AzdElDZ2J5QjU5dno3U2s0TkFtak1ldjlhb0FvcE1wRWZ6NWcxcEY1cXdKcnV2bG9MQkFoQUROdw&q=https%3A%2F%2Fstatweb.stanford.edu%2F%7Ejhf%2Fftp%2Fstobst.pdf&v=3CC4N4z3GJc) * Tài liệu về XGB tại [đây](https://arxiv.org/pdf/1603.02754.pdf) * Ngoài ra nhóm còn tham khảo trên nhiều nguồn khác  1. **Goals And Deliverables**   Kế hoạch và các mốc cần đạt được dự kiến:   * Thực hiện các quy trình lập trình tuần tự cho thuật toán, đồng thời tiến hành kiểm thử thuật toán đã cài đặt cho một lượng sample đã có để xác nhận kết quả lần đầu (độ chính xác, thời gian khởi chạy tốn?) → hoàn thành mức độ 50% * Tiến hành tách riêng các phần trong thuật toán Xgboost tiến hành cài đặt song song hóa bằng numba (Tối ưu phần tính numpy cho các vectơ hóa, Numba tối ưu vòng lặp, đặc biệt ở phần xây dựng cây và thuật toán tối ưu hóa hàm mất mát) → độ chính xác của mô hình tăng | hoàn thành mức độ 80% * Bắt đầu huấn luyện mô hình ở mức độ cơ bản (với dữ liệu ít, có sẵn nhãn), với mục đích kiểm tra hiệu xuất mô hình song song hóa mang lại đồng thời so sánh với thuật toán tuần tự đã cài đặt (Phần nào đã cải thiện, tốc độ chạy có nhanh hơn không?) → hoàn thành mức độ 100% * Với việc áp dụng phần dữ liệu lớn, dữ liệu không gán nhãn và các đặc trưng khác (features) là một thử thách cho mô hình XGBoost đã được song song hóa, điều này có thể ứng dụng ít nhiều vào thực tế đối với đầu vào là ảnh thì ta có thể trả kết quả là tính trạng bệnh (cung cấp thêm các đặc trưng quan trọng, chọn lọc đặc trưng) → hoàn thành mức độ 130%.   Phần trình bày cho đồ án cuối kỳ   * Với mức độ là đồ án với nhiệm vụ chính là việc song song hóa Xgboost cho việc nhận dạng bệnh trên lá, sẽ đưa ra một số hình thức có thể trình bày của nhóm cho thầy và các bạn trong buổi seminar như sau: * Demo mô tả: có thể thực hiện với hình ảnh của một chiếc lá làm đầu vào và mô hình Xgboost song song hóa sẽ dự đoán bệnh đang ảnh hưởng trên chiếc là đó (trên thời gian thực và mức độ chính xác của mô hình nhóm xây dựng). * Xuất kết quả đầu ra: có thể làm nổi bật vùng bị bệnh của chiếc lá đó cùng với các thông tin về bệnh tình của chiếc lá đó (tính hiểu quả trong nhận dạng và khả năng phân loại các loại bệnh trên lá). * Trình bày đồ thị: có thể so sánh thời gian thực thi, hiệu suất giữa 2 mô hình tuần tự và song song (việc cài đặt song song hóa có lợi ích trong việc tiết kiệm thời gian tính toán) * Đánh giá độ chính xác: Thông các đồ thị có trên thì có kết quả cho việc phân loại bệnh cùng mức độ chính xác ở cả 2 mô hình (đặc biệt nếu mô hình song song hóa tiết kiệm thời gian tính toán cũng như đạt được độ chính xác cao). * Khả năng mở rộng: đối với việc xử lý với các kích thước dữ liệu khác nhau, đặc biệt đối với dữ liệu lớn. * Cuối cùng, áp dụng với các kỹ thuật song song hóa khác nhau có thể dụng sử dụng đối với Xgboost, cùng với mặt lợi và hại của mỗi phương pháp đó và giải thích.   Kiến thức đạt được trong việc nghiên cứu đồ án này:   * Về mức độ là đồ án nghiên cứu thì có thể đưa ra một số thông tin về thông số hệ thống sử dụng (thời gian xử lý, tài nguyên máy tính được sử dụng), tối ưu hiệu suất (tìm kiếm các cải tiến và tối ưu hóa để nâng cao hiệu suất của mô hình XGBoost. Đối với việc cài đặt các cấu hình tham số (parameters) tối ưu, phương pháp song song hóa, hoặc tối ưu hóa khác để giảm thời gian xử lý hoặc tăng độ chính xác của mô hình.), tính mở rộng (khả năng mở rộng của hệ thống khi xử lý dữ liệu lớn, từ đó có thể đánh giá hiệu suất và khả năng xử lý của mô hình XGBoost trên các tập dữ liệu có kích thước khác nhau để hiểu cách hệ thống xử lý tải trọng công việc tăng lên.), độ chính xác và mức độ tin cậy (phân tích về 2 mức độ của mô hình XGBoost trong việc nhận dạng bệnh trên lá cây. Và có thể đo lường độ chính xác của mô hình và đánh giá khả năng dự đoán chính xác các loại bệnh khác nhau.), cuối cùng là phần kết luận tổng quan về hệ thống cũng như mô hình sử dụng (hiểu về cấu trúc cũng như hệ thống của mô hình Xgboost hoạt động khi áp dụng vào bài toán nhận dạng bệnh trên lá, cũng như có thể phân tích thêm cách dữ liệu được chia nhỏ, sử dụng bộ nhớ và tương tác giữa các thành phần của hệ thống). * Cần tìm hiểu sâu, chi tiết hơn về bài toán mà nhóm đang xử lý cũng như đọc thêm các tài liệu nghiên cứu khoa học của những người tiền nhiệm để từ đó cải biến, cải thiện thêm hiệu suất của mô hình Xgboost trong bài toán nói trên.   Mức độ hiểu và cài đặt hệ thống cho đồ án:   * Hệ thống tồn tại : Khả năng của hệ thống (hệ thống mà nhóm có thể sử dụng cũng như có khả năng thực hiện về nhận dạng và phân loại bệnh trên lá, khả năng xử lý ảnh, khả năng tích hợp với các công cụ và giao diện khác, và khả năng tương tác với người dùng ), hiệu suất dự kiến (thời gian xử lý, độ chính xác, tài nguyên máy tính sử dụng (bộ nhớ, CPU), và khả năng xử lý tải công việc), mục tiêu hiệu suất (xác định các tiêu chí đánh giá hệ thống sử dụng của nhóm như độ chính xác, thời gian phản hồi, tốc độ xử lý, tài nguyên sử dụng, và các yếu tố khác phù hợp với bài toán), các tiêu chí đánh giá, các tính năng và các cải tiến khác (khả năng tích hợp các tính khác của hệ thống mà nhóm có thể sử dụng) |
| --- |

| **Weekly schedule:** Kế hoạch dự kiến   | Tuần | Người thực hiện | Tên công việc | Mô tả | | --- | --- | --- | --- | | Week 01 | Quốc Anh ,Duy, Nam Anh | Tìm tài liệu cho bài toán | Tìm kiếm các tài liệu, chủ đề, hướng dẫn cài đặt liên quan phần lớn đến XGB và về bài toán nhận dạng bệnh trên lá. | | Week 02-03 | Quốc Anh ,Duy, Nam Anh | Code tuần tự | Tiến hành code tuần tự chi tiết cho thuật toán XGB bằng thư viện numpy | | Week 04 | Quốc Anh, Duy, Nam Anh | Kiểm thử mô hình tuần tự | Đối với thuật toán tuần tự mà nhóm đã code | | Week 05-06 | Quốc Anh, Duy, Nam Anh | Song song hóa thuật toán | Tiến hành phân tích cũng như song song hóa, cải thiện phần mô hình tuần tự với mục tiêu tính toán thời gian nhanh hơn, hiệu suất mô hình tốt hơn | | Week 07 | Quốc Anh, Duy, Nam Anh | Kiểm thử mô hình song song | Với dữ liệu đã có để so sánh với kết quả của code tuần tự | | Week 08 | Quốc Anh, Duy, Nam Anh | Mở rộng | Về dữ liệu lớn và phương pháp xử lý dữ liệu | | Week 09 - 10 | Quốc Anh, Duy, Nam Anh | Điều chỉnh phần mở rộng | Có thể áp dụng được nhiều hệ thống hơn không, cần chỉnh sửa phần song song hóa để mô hình đạt hiệu suất cao hơn? | | Week 10 - 11 | Quốc Anh, Duy, Nam Anh | Viết báo cáo | Viết báo cáo về đồ án của nhóm thực hiện cũng như phân tích nhóm đưa ra | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |