**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**
**KHOA CÔNG NGHỆ THÔNG TIN**


# BÁO CÁO TASK 2

## KHAI THÁC DỮ LIỆU ĐỒ THỊ

### Đồ án cuối kỳ - 21KHDL

**Nhóm 7 - Sinh viên thực hiện:**

21127104 - Đoàn Ngọc Mai

21127129 - Lê Nguyễn Kiều Oanh

21127229 - Dương Trường Bình

21127616 - Lê Phước Quang Huy

**Giảng viên hướng dẫn:**

Lê Nhựt Nam

# Mục lục


1 Thông tin nhóm . . . . . 3

2 Current Status . . . . . 4

2.1 Tiến độ công việc . . . . . 4

2.2 Khó khăn gặp phải và Cách giải quyết . . . . . 4

3 Introduction . . . . . 6

3.1 Bối cảnh và vấn đề . . . . . 6

3.2 Vấn đề . . . . . 6

3.3 Động lực . . . . . 7

3.4 Tiềm Năng . . . . . 7

3.5 Ý Nghĩa Khoa Học . . . . . 7

3.6 Ý Nghĩa Thực Tế và Ứng Dụng . . . . . 8

4 Preliminaries and Backgrounds . . . . . 9

4.1 Ký Hiệu và Định Nghĩa . . . . . 9

4.2 Phát Biểu Bài Toán . . . . . 11

4.3 Framework chung để giải quyết bài toán . . . . . 11

4.4 Thách Thức và Hạn Chế . . . . . 12

5 Related Works . . . . . 13

5.1 Diễn Tiến Phát Triển Bài Toán . . . . . 13

5.2 Khoảng Trống Nghiên Cứu . . . . . 14

5.3 Timeline Phát Triển . . . . . 15

5.4 Công Trình Dựa Trên . . . . . 17
6 Methodology . . . . . 19
6.1 Hypergraph Neural Network (HNN) . . . . . 19
6.2 Input và Output . . . . . 20
6.3 Tổng quan về ViHGNN Framework . . . . . 21
6.4 Cấu trúc Hpergraph của hình ảnh . . . . . 21
6.5 Tối ưu hóa cấu trúc hypergraph thích ứng và tối ưu hóa end-to-end . . . . . 22
6.6 Chi tiết cấu hình của ViHGNN . . . . . 22
6.7 Lợi ích và Bất lợi . . . . . 25
7 Experimental and Evaluations . . . . . 26
7.1 Quy Trình Thực Hiện . . . . . 26
7.2 Kết Quả . . . . . 28
8 Kết Luận . . . . . 39
Tài liệu tham khảo . . . . . 43

# 1 Thông tin nhóm

| MSSV | Họ và tên | Công việc được phân công | Mức độ hoàn thành |
|---|---|---|---|
| 21127104 | Đoàn Ngọc Mai | • Đọc paper Vision HGNN và tìm hiểu về phần Introduction, Preliminaries and Backgrounds và Methodology.<br>• Chuẩn bị slide thuyết trình về các nội dung đã tìm hiểu.<br>• Viết báo cáo.<br>• Thuyết trình buổi Seminar. | 100% |
| 21127129 | Lê Nguyễn Kiều Oanh | • Đọc paper Vision HGNN và tìm hiểu về phần Related Works, Experimental & Evaluations và Conclusion.<br>• Chuẩn bị slide thuyết trình về nội dung liên quan.<br>• Viết báo cáo.<br>• Thuyết trình buổi Seminar. | 100% |
| 21127229 | Dương Trường Bình | • Đọc paper Vision GNN và tìm hiểu về phần Related Works, Experimental & Evaluations và Conclusion.<br>• Kiểm thử lại code về GNN và xem xét kết quả kiểm thử.<br>• Viết báo cáo.<br>• Chuẩn bị slide thuyết trình về các nội dung liên quan..<br>• Thuyết trình buổi Seminar. | 100% |
| 21127616 | Lê Phước Quang Huy | • Đọc paper Vision GNN và tìm hiểu về phần Introduction, Preliminaries and Backgrounds và Methodology.<br>• Viết báo cáo.<br>• Chuẩn bị slide thuyết trình về các nội dung đã tìm hiểu.<br>• Thuyết trình buổi Seminar. | 100% |

# 2 Current Status

## 2.1 Tiến độ công việc

| MSSV | Công việc | Thời gian thực hiện | Trạng thái công việc |
|---|---|---|---|
| 21127104 | Đọc paper Vision HGNN và tìm hiểu về phần Introduction, Preliminaries and Backgrounds và Methodology. | 2 tuần | Đã hoàn thành |
| | Chuẩn bị slide thuyết trình về các nội dung đã tìm hiểu. | 1 tuần | Đã hoàn thành |
| | Viết báo cáo. | 1 tuần | Đã hoàn thành |
| | Thuyết trình buổi Seminar. | 1 buổi | Đã hoàn thành |
| 21127129 | Đọc paper Vision HGNN và tìm hiểu về phần Related Works, Experimental & Evaluations và Conclusion. | 2 tuần | Đã hoàn thành |
| | Thực nghiệm code paper Vision HGNN. | 1 tuần | Chưa hoàn thành |
| | Chuẩn bị slide thuyết trình về nội dung liên quan. | 1 tuần | Đã hoàn thành |
| | Viết báo cáo. | 1 tuần | Đã hoàn thành |
| | Thuyết trình buổi Seminar. | 1 buổi | Đã hoàn thành |
| 21127229 | Đọc paper Vision HGNN và tìm hiểu về phần Related Works, Experimental & Evaluations và Conclusion. | 2 tuần | Đã hoàn thành |
| | Thực nghiệm code về Vision GNN và kiểm tra kết quả. | 1 tuần | Đã hoàn thành |
| | Viết báo cáo. | 1 tuần | Đã hoàn thành |
| | Chuẩn bị slide thuyết trình về các nội dung liên quan. | 1 tuần | Đã hoàn thành |
| | Thuyết trình buổi Seminar. | 1 buổi | Đã hoàn thành |
| 21127616 | Đọc paper Vision GNN và tìm hiểu về phần Introduction, Preliminaries and Backgrounds và Methodology. | 2 tuần | Đã hoàn thành |
| | Chuẩn bị slide thuyết trình về các nội dung đã tìm hiểu. | 1 tuần | Đã hoàn thành |
| | Viết báo cáo. | 1 tuần | Đã hoàn thành |
| | Thuyết trình buổi Seminar. | 1 buổi | Đã hoàn thành |

## 2.2 Khó khăn gặp phải và Cách giải quyết

- **Khó khăn:** Khối lượng tài liệu lớn và độ phức tạp cao.
  - **Cách giải quyết:** Chia nhỏ công việc, phân công nhiệm vụ cho từng thành viên trong nhóm

theo các phần cụ thể của bài báo. Điều này giúp tăng cường sự tập trung vào từng phần nội dung, đồng thời giảm áp lực tổng thể.

- **Khó khăn:** Khó khăn trong việc hiểu và áp dụng các khái niệm phức tạp về HGNN (Hypergraph Neural Networks).
  - **Cách giải quyết:** Tổ chức các buổi thảo luận nhóm để trao đổi và giải thích các khái niệm phức tạp. Ngoài ra, tham khảo thêm các tài liệu phụ trợ như sách, bài giảng trực tuyến, và video giải thích về HGNN để có cái nhìn toàn diện hơn.
- **Khó khăn:** Thời gian hạn chế dẫn đến áp lực về tiến độ hoàn thành công việc.
  - **Cách giải quyết:** Lập kế hoạch cụ thể và tuân thủ chặt chẽ thời gian đã đặt ra. Ưu tiên các công việc quan trọng và tránh lãng phí thời gian vào những nhiệm vụ không cần thiết.
- **Khó khăn:** Gặp nhiều trở ngại trong quá trình thực nghiệm cho cả Vision GNN và Vision HGNN do giới hạn về tài nguyên và các vấn đề kỹ thuật.
  - **Đối với Vision GNN:** Bộ dữ liệu ImageNet quá lớn để lưu trữ trên máy cá nhân, mã nguồn từ GitHub (2022) có các phiên bản thư viện cũ gây xung đột, không có mô hình pretrained của kiến trúc Pyramid và thiếu phần cứng phù hợp để huấn luyện mô hình.
  - **Đối với Vision HGNN:** Gặp vấn đề tương tự về bộ dữ liệu, đồng thời mã nguồn trên GitHub còn nhiều lỗi và không thể chạy được.
  - **Cách giải quyết:**
    - Đối với Vision GNN: Nhóm đã sử dụng bộ dữ liệu nhỏ hơn (Tiny ImageNet 200) và chỉ sử dụng mô hình pre-trained cho kiến trúc isotropic. Nhóm đã khắc phục các lỗi thư viện và thực hiện inference thay vì huấn luyện mô hình.
    - Đối với Vision HGNN: Do các vấn đề kỹ thuật phức tạp, nhóm chưa thể thực hiện được phần thực nghiệm.

# 3 Introduction

## 3.1 Bối cảnh và vấn đề

Những bước tiến nhanh chóng trong học sâu đã mang lại những thành công đáng kể trong các mô hình thị giác máy tính đa dạng. Chúng bao gồm Mạng Nơ-ron Tích chập (CNNs), Vision Transformers (ViTs) và các mô hình thị giác dựa trên MLP. Mặc dù đã có những thành công đáng kể với những công nghệ này, tiềm năng của cấu trúc đồ thị trong xử lý hình ảnh vẫn chưa được khai thác đầy đủ. ViG đã mở ra một hướng mới bằng cách sử dụng GNNs và phân chia hình ảnh thành các mảnh để giảm số lượng node. Tuy nhiên, ViG vẫn gặp phải một số hạn chế, như không thể mô hình hóa các mối quan hệ phức tạp và tạo ra các cạnh dư thừa. Để giải quyết những vấn đề này, tác giả đề xuất một mô hình mới gọi là ViHGNN, sử dụng siêu đồ thị để biểu diễn hình ảnh. Siêu đồ thị có khả năng liên kết nhiều đỉnh cùng lúc, giúp mô hình hóa các mối quan hệ phức tạp tốt hơn. Họ đã sử dụng phương pháp Fuzzy C-Means để xây dựng và cập nhật siêu đồ thị một cách hiệu quả và ít tốn kém. Kết quả cho thấy mô hình ViHGNN vượt trội trong các nhiệm vụ phân loại hình ảnh và nhận diện đối tượng so với các mô hình trước đây.

## 3.2 Vấn đề

- **Giới hạn của đồ thị đơn giản:** Đồ thị đơn giản chỉ có thể kết nối các cặp nút, không thể nắm bắt các mối quan hệ phức tạp giữa nhiều nút cùng một lúc.
- **Bộ nhớ và chi phí tính toán cao:** Khi tăng số lượng nút và cạnh để nắm bắt nhiều thông tin hơn, chi phí bộ nhớ và tính toán của các mô hình đồ thị đơn giản tăng lên đáng kể.
- **Hiệu suất biểu diễn:** Khả năng biểu diễn và học các mối quan hệ phức tạp trong hình ảnh của các mô hình đồ thị đơn giản không cao, dẫn đến hiệu suất thấp trong các nhiệm vụ thị giác máy tính như phân loại hình ảnh và phát hiện đối tượng.

### 3.3 Động lực

Các phương pháp trước đây như ViG (Vision Graph Neural Network) đã sử dụng đồ thị để xử lý hình ảnh bằng cách chia hình ảnh thành các mảnh (patch) và tạo kết nối giữa các mảnh lân cận. Tuy nhiên, phương pháp này chỉ xử lý được các mối quan hệ đôi (pairwise), dẫn đến việc tạo ra quá nhiều cạnh không cần thiết và tăng chi phí tính toán. Việc sử dụng siêu đồ thị trong ViHGNN giúp khắc phục những hạn chế này, nắm bắt các mối quan hệ bậc cao hơn trong hình ảnh và giảm chi phí tính toán.

### 3.4 Tiềm Năng

- **Tăng cường hiệu quả xử lý hình ảnh:** Việc sử dụng mạng HyperGraph Neural Network (HGNN) để đại diện hình ảnh mang lại tiềm năng nắm bắt các mối quan hệ phức tạp giữa các patch của hình ảnh, điều mà các mô hình truyền thống như CNN và ViT khó làm được.
- **Cải thiện hiệu suất trong các tác vụ thị giác:** ViHGNN đã được chứng minh là cải thiện hiệu suất trong các tác vụ phân loại hình ảnh và phát hiện đối tượng, với độ chính xác top-1 đạt 83.9% trong tác vụ phân loại ImageNet và 43.1% Average Precision (AP) trong tác vụ phát hiện đối tượng COCO.

### 3.5 Ý Nghĩa Khoa Học

- **Đóng góp vào lý thuyết về mạng neural đồ thị:** Việc mở rộng từ Graph Neural Networks (GNNs) sang HyperGraph Neural Networks (HGNNs) giúp nắm bắt tốt hơn các mối quan hệ bậc cao trong dữ liệu hình ảnh, đồng thời đề xuất một cấu trúc đồ thị siêu mới mẻ và hiệu quả hơn.
- **Tiến bộ trong học cấu trúc đồ thị:** Việc áp dụng phương pháp Fuzzy C-Means để xây dựng và cập nhật cấu trúc hypergraph trong quá trình huấn luyện và suy diễn giúp giảm tải tính toán và tăng độ chính xác.

### 3.6 Ý Nghĩa Thực Tế và Ứng Dụng

- **Ứng dụng trong phân loại và phát hiện đối tượng:** ViHGNN có thể được ứng dụng trong các hệ thống phân loại hình ảnh và phát hiện đối tượng, từ đó cải thiện độ chính xác và hiệu suất của các hệ thống thị giác máy tính.
- **Tối ưu hóa các hệ thống thị giác trong thời gian thực:** Việc giảm bớt tính toán và bộ nhớ thông qua việc tối ưu hóa cấu trúc hypergraph giúp các hệ thống thị giác thời gian thực hoạt động hiệu quả hơn.

# 4 Preliminaries and Backgrounds

## 4.1 Ký Hiệu và Định Nghĩa

- **Data Matrix:** Tập hợp các đoạn ảnh, mỗi đoạn được biểu diễn bởi một vector đặc trưng. Giả sử hình ảnh có $N$ đoạn ảnh, và mỗi đoạn được biểu diễn bởi một vector có chiều $D$. Ma trận dữ liệu này có thể được ký hiệu là $X$, bao gồm $N$ cột và mỗi cột là vector đặc trưng của một đoạn ảnh.
- **Hypergraph:** Hypergraph là một cấu trúc toán học bao gồm các đỉnh và hyperedge, nơi mỗi hyperedge có thể kết nối nhiều đỉnh. Hypergraph được biểu diễn bằng ma trận incidence $H \in \{0,1\}^{N\times E}$, trong đó $H_{ie} = 1$ nếu hyperedge $e$ chứa đỉnh $v_i$, và $H_{ie} = 0$ nếu không. Bậc của đỉnh và hyperedge lần lượt được xác định bởi $D_{ii} = \sum_{e=1}^{E} H_{ie}$ và $B_{ee} = \sum_{i=1}^{N} H_{ie}$. Ma trận chéo $D$ và $B$ chứa các bậc của đỉnh và hyperedge trên đường chéo của chúng.
- **Degree of Node:** Số lượng siêu cạnh kết nối với một đỉnh cụ thể.
- **Degree of Hyperedge:** Số lượng đỉnh mà siêu cạnh đó kết nối.
- **HGNN - Hypergraph Neural Network:** Một loại mạng neural được thiết kế để làm việc với hypergraph. Nó bao gồm các lớp đặc biệt để xử lý thông tin từ các siêu cạnh và cập nhật đặc trưng của các đỉnh. Mỗi lớp trong HGNN có thể được hiểu như sau:
  - **Đầu vào (Input):** Đặc trưng của các đỉnh từ lớp trước.
  - **Xử lý (Processing):** Sử dụng siêu cạnh để truyền thông tin và cập nhật đặc trưng của các đỉnh.
  - **Đầu ra (Output):** Đặc trưng mới của các đỉnh sau khi đã được cập nhật.
- **Fuzzy C-Means:** Một phương pháp phân cụm được sử dụng để xây dựng và cập nhật hypergraph. Phương pháp này giúp chia các đỉnh thành các cụm, trong đó mỗi đỉnh có thể thuộc nhiều cụm với các mức độ khác nhau (fuzzy).

- **Isotropic ViHGNN:** Mạng Neural Hypergraph Vision (ViHGNN) đẳng hướng. "Isotropic"nghĩa là tính đồng nhất theo mọi hướng, tức là các đặc trưng xử lý bởi mạng có cùng cấu trúc và phương pháp trong mọi phần của mạng.
- **FFN (Feed-Forward Network):** Mạng truyền thẳng. Đây là một loại mạng neural đơn giản, nơi các tín hiệu di chuyển theo một chiều từ lớp đầu vào đến lớp đầu ra qua các lớp ẩn mà không có bất kỳ vòng lặp nào.
- **FLOPs (Floating Point Operations):** Số lượng phép toán dấu chấm động. Đây là một đơn vị đo lường hiệu suất của các mô hình học máy, chỉ số lượng phép toán dấu chấm động cần thiết để thực hiện một thuật toán hoặc một mô hình.
- **Pyramid ViHGNN:** Mạng Neural Hypergraph Vision (ViHGNN) dạng kim tự tháp. Mô hình này tổ chức các lớp mạng theo dạng kim tự tháp, bắt đầu từ các lớp với độ phân giải cao ở đáy và giảm dần độ phân giải khi lên đỉnh, giúp tổng hợp thông tin từ nhiều mức độ phân giải khác nhau.
- **Isotropic networks:** Mạng đẳng hướng. Đây là các mạng neural có cấu trúc đồng nhất, xử lý thông tin theo cùng một cách trong toàn bộ mạng, không phân biệt các phần khác nhau của dữ liệu đầu vào.
- **Pyramid networks:** Mạng dạng kim tự tháp. Đây là các mạng neural tổ chức theo cấu trúc kim tự tháp, với các tầng giảm dần độ phân giải từ đáy lên đỉnh, giúp tổng hợp và trích xuất thông tin từ các mức độ phân giải khác nhau của dữ liệu đầu vào.
- **Hyper-parameters:** Siêu tham số. Đây là các tham số cấu hình bên ngoài được thiết lập trước khi huấn luyện mô hình học máy. Siêu tham số ảnh hưởng đến quá trình học của mô hình, ví dụ như tỷ lệ học (learning rate), số lượng lớp (number of layers), kích thước batch (batch size), v.v. Siêu tham số khác với tham số mô hình, là những giá trị được học trong quá trình huấn luyện.

### 4.2 Phát Biểu Bài Toán

Bài báo giải quyết vấn đề chính là cách đại diện hình ảnh một cách hiệu quả để mô hình hóa các quan hệ phức tạp trong hình ảnh. Các phương pháp trước đây như CNNs, ViTs và ViG đều có hạn chế trong việc biểu diễn các quan hệ bậc cao và tạo ra các cạnh dư thừa trong đồ thị. Việc sử dụng đồ thị đơn giản chỉ cho phép kết nối cặp đôi, không phù hợp với các quan hệ phức tạp giữa các mảnh của hình ảnh. Vì vậy, bài báo đề xuất sử dụng siêu đồ thị để giải quyết vấn đề này .

### 4.3 Framework chung để giải quyết bài toán

Bài toán chính được đề cập trong bài báo là tìm cách cải thiện hiệu suất của các mô hình thị giác máy tính trong các tác vụ như phân loại ảnh và phát hiện đối tượng. Framework chung để giải quyết bài toán này thường bao gồm các bước sau:

- **Biểu diễn ảnh:** Chuyển đổi ảnh đầu vào thành một dạng biểu diễn phù hợp cho việc xử lý.
  - **ViHGNN:** Chia ảnh thành các patch nhỏ và biểu diễn dưới dạng các embedding (nhúng) patch. Sau đó, các patch này được gán vào các đỉnh trong một hypergraph.
- **Xây dựng Hypergraph:** Sử dụng phương pháp Fuzzy C-Means để xây dựng và cập nhật cấu trúc của hypergraph từ các patch embedding.
  - **Hypergraph:** Các đỉnh trong hypergraph đại diện cho các patch embedding, và các cạnh hyperedge biểu thị các mối quan hệ đa chiều giữa các đỉnh.
- **Trích xuất đặc trưng:** Sử dụng các mô-đun của ViHGNN để học và trích xuất các đặc trưng từ hypergraph.
  - **HGNN (Hypergraph Neural Network):** Sử dụng các lớp tích chập trên hypergraph để học các đặc trưng của các patch embedding.
  - **FFN (Feed-Forward Network):** Một mạng truyền thẳng được sử dụng sau các lớp HGNN để tiếp tục xử lý và tối ưu hóa các đặc trưng đã học.

- **Tổng hợp thông tin:** Kết hợp thông tin từ các đặc trưng đã trích xuất từ hypergraph.
  - **ViHGNN:** Sử dụng nhiều block ViHGNN để cập nhật và tổng hợp thông tin từ các đặc trưng đã trích xuất, tạo ra các embedding patch được cập nhật.
- **Phân loại hoặc dự đoán:** Sử dụng các lớp fully connected hoặc MLP để đưa ra kết quả cuối cùng.
  - **Output Head:** Kết hợp thông tin từ các embedding patch đã cập nhật để đưa ra dự đoán cuối cùng trong tác vụ phân loại hoặc nhận diện đối tượng.

Với sự kết hợp của hypergraph và các mô hình neural network, ViHGNN cung cấp một cách tiếp cận mới nhằm tăng cường hiệu suất và khả năng xử lý thông tin từ dữ liệu ảnh, giúp giải quyết tốt hơn các bài toán thị giác máy tính.

### 4.4 Thách Thức và Hạn Chế

Một trong những thách thức lớn nhất trong việc sử dụng siêu đồ thị để đại diện cho hình ảnh là làm sao có thể xác định cấu trúc siêu đồ thị tối ưu cho việc biểu diễn hình ảnh. Việc xây dựng một cấu trúc siêu đồ thị hiệu quả đòi hỏi phải xử lý được vấn đề "con gà và quả trứng", tức là việc cần có siêu đồ thị để đại diện cho hình ảnh, trong khi cấu trúc siêu đồ thị lại chưa có sẵn. Quá trình này phụ thuộc nhiều vào các embedding của các patch, và các embedding này lại phụ thuộc vào cấu trúc siêu đồ thị, tạo ra một vòng lặp phản hồi phức tạp.

Hơn nữa, việc xây dựng và cập nhật cấu trúc siêu đồ thị yêu cầu sự cẩn thận để tránh những kết nối không cần thiết, có thể dẫn đến việc giảm hiệu suất của mô hình. Các thông tin nhiễu có thể làm giảm khả năng phân biệt hình ảnh và ảnh hưởng tiêu cực đến các tác vụ phân loại và nhận diện đối tượng. Do đó, tối ưu hóa cấu trúc siêu đồ thị trở thành một yêu cầu cần thiết để đảm bảo mô hình có thể đạt được hiệu suất tốt nhất có thể. Tuy nhiên, vấn đề này vẫn còn đang mở và cần được nghiên cứu thêm trong tương lai để cải thiện.

# 5 Related Works

## 5.1 Diễn Tiến Phát Triển Bài Toán

- **Thuở ban đầu**
  - **CNNs trong thị giác máy tính:** Ban đầu, CNNs là phương pháp chủ đạo cho các nhiệm vụ thị giác máy tính như phân loại hình ảnh, nhận diện đối tượng và phân đoạn ngữ nghĩa. Các mạng nổi bật như LeNet (1998), AlexNet (2012), VGGNet (2014) và ResNet (2015) đã đóng vai trò quan trọng trong việc cải thiện hiệu suất và độ chính xác của các mô hình thị giác máy tính.
  - **GNNs trong mạng xã hội và sinh hóa:** GNNs ban đầu được áp dụng trong các lĩnh vực như mạng xã hội, mạng trích dẫn và đồ thị sinh hóa. Các công trình này tập trung vào việc xử lý các dữ liệu phi cấu trúc và mối quan hệ phức tạp giữa các nút trong đồ thị.
- **Gần đây**
  - **ViTs và MLPs trong thị giác máy tính:** Với sự thành công của Transformer trong NLP, Vision Transformer (ViTs) được giới thiệu vào năm 2020 để áp dụng kiến trúc này cho các nhiệm vụ thị giác. Các biến thể của ViTs cải thiện hiệu suất thông qua kiến trúc kim tự tháp, sự chú ý địa phương và mã hóa vị trí. Ngoài ra, các kiến trúc MLP cũng được khám phá để thay thế cho các CNNs và ViTs.
  - **GNNs và HGNNs trong thị giác máy tính:** GNNs bắt đầu được áp dụng rộng rãi trong thị giác máy tính cho các nhiệm vụ như phân loại và phân đoạn đám mây điểm, tạo đồ thị cảnh và nhận diện hành động con người. Siêu đồ thị (HGNNs) được sử dụng trong các nhiệm vụ như tìm kiếm hình ảnh, phân loại đối tượng 3D và nhận diện lại người, giúp nắm bắt các mối quan hệ phức tạp hơn so với đồ thị thông thường.
- **Phát triển qua thời gian**
  - **Cải tiến CNNs:** Các mạng như ResNet, MobileNet và các mạng tìm kiếm bởi NAS đã nâng

cao khả năng học và giảm thiểu độ phức tạp của mô hình.

- **Ứng dụng ViTs:** ViT và các biến thể đã mở rộng khả năng của Transformer sang lĩnh vực thị giác máy tính, giúp cải thiện hiệu suất của các nhiệm vụ phân loại và nhận diện đối tượng.
- **Học cấu trúc đồ thị và siêu đồ thị:** Các phương pháp học cấu trúc đồ thị và siêu đồ thị đã được phát triển để tối ưu hóa việc biểu diễn và xử lý dữ liệu đồ thị, mặc dù vẫn còn nhiều thách thức về chi phí tính toán và khả năng hội tụ.

## 5.2 Khoảng Trống Nghiên Cứu

Bài báo đã xác định được một khoảng trống nghiên cứu trong lĩnh vực biểu diễn hình ảnh bằng cách sử dụng dữ liệu siêu đồ thị và mạng thần kinh siêu đồ thị cho các nhiệm vụ thị giác. Mặc dù các ạng GNN và Mạng HGNN đã được sử dụng rộng rãi trong thị giác máy tính cho các nhiệm vụ như phân loại và phân đoạn đám mây điểm, tạo đồ thị cảnh và nhận dạng hành động của con người, nhưng việc áp dụng chúng để xử lý trực tiếp dữ liệu hình ảnh vẫn còn hạn chế.

Tác giả bài báo đã nhấn mạnh rằng trong khi ViG là một trong số ít phương pháp xử lý trực tiếp dữ liệu hình ảnh, nhưng việc tích hợp các siêu đồ thị để biểu diễn hình ảnh vẫn còn là một thách thức.

Cụ thể hơn, bài báo đã giải quyết hai hạn chế chính của việc sử dụng các đồ thị đơn giản để biểu diễn hình ảnh, đó là lý do tại sao họ đề xuất siêu đồ thị.

Do đó, sự mới lạ của bài báo nằm ở việc giới thiệu ViHGNN, một mô hình biểu diễn hình ảnh dưới dạng siêu đồ thị động. Điều này này vượt qua những hạn chế của các phương pháp dựa trên đồ thị trước đó bằng cách nắm bắt các mối quan hệ bậc cao trong khi giảm thiểu chi phí bộ nhớ và tính toán dư thừa.

### 5.3 Timeline Phát Triển

Dưới đây là timeline phát triển của các phương pháp trong lĩnh vực thị giác máy tính, từ các mô hình truyền thống đến các phương pháp tiên tiến như ViHGNN:

- **Trước 2012: Phương pháp truyền thống**
  - **Các phương pháp dựa trên đặc trưng thủ công:** SIFT-1999 (Scale-Invariant Feature Transform), SURF-2006 (Speeded-Up Robust Features), HOG-2005 (Histogram of Oriented Gradients)
  - **Nhược điểm:** Hiệu suất thấp, khó khăn trong việc xử lý các hình ảnh phức tạp.
- **2012: Sự ra đời của CNN (Convolutional Neural Networks)**
  - **AlexNet (2012):** Mô hình CNN đầu tiên đạt hiệu suất cao trên ImageNet, khởi đầu cho sự bùng nổ của CNN trong thị giác máy tính.
  - **Đặc điểm:** Sử dụng các lớp convolution để trích xuất đặc trưng không gian trong ảnh.
- **2012 - 2017:**
  - Mạng CNNs tiếp tục thống trị thị giác máy tính, với các kiến trúc mới được phát triển để cải thiện hiệu suất như
    - **VGGNet (2014):** Sử dụng các lớp convolution nhỏ và sâu hơn.
    - **ResNet (2015):** Giới thiệu khái niệm residual connections, cho phép xây dựng các mô hình cực sâu.
    - **Inception (2015):** Sử dụng các kiến trúc convolution phức tạp để cải thiện hiệu suất và giảm thiểu chi phí tính toán.
  - **Mạng đồ thị GNNs**, mặc dù trước đó thường được sử dụng trong các lĩnh vực như mạng xã hội và mạng lưới trích dẫn, đã bắt đầu được khám phá cho các tác vụ thị giác máy tính như phân loại và phân đoạn đám mây điểm

    - **Đặc điểm:** Sử dụng các nút để đại diện cho các vùng trong ảnh và các cạnh để biểu diễn mối quan hệ giữa chúng

  - **2017:** Transformer thành công trong NLP, khơi nguồn cảm hứng cho Vision Transformer

- **2018:**

  - **Mạng siêu đồ thị (HGNN)** bắt đầu thu hút sự chú ý cho các tác vụ thị giác máy tính như truy xuất hình ảnh và phân loại đối tượng 3D, tận dụng khả năng nắm bắt mối quan hệ phức tạp giữa dữ liệu.

- **2018 - 2020:**

  - **Nghiên cứu về học cấu trúc đồ thị / siêu đồ thị** đã tăng lên, nhằm mục đích học cấu trúc đồ thị / siêu đồ thị tối ưu để cải thiện hiệu suất của GNN và HGNN.

  - **2020: Vision Transformer (ViT)** được giới thiệu và ra mắt, áp dụng kiến trúc Transformer cho các tác vụ thị giác, mở ra một hướng nghiên cứu mới.

    - **Đặc điểm:** ViT chia hình ảnh thành các patch và sử dụng các lớp self-attention để học các đặc trưng của ảnh

    - **Ưu điểm:** ViT có khả năng nắm bắt mối quan hệ dài hạn giữa các vùng của ảnh, đạt được hiệu suất cao trên nhiều bộ dữ liệu

    - **Nhược điểm:** ViT yêu cầu lượng dữ liệu lớn và tài nguyên tính toán mạnh để huấn luyện

- **2021 - 2023**

  - Nhiều biến thể của **ViT** đã được đề xuất để cải thiện hiệu suất

  - **2021 - 2022:** Kiến trúc MLP cũng được khám phá trong thị giác máy tính, bao gồm ConvMixer (2021), ResMLP (2021), CycleMLP (2022), AS-MLP (2022), Hire-MLP (2022)

    - Sự ra đời của MLP-Mixer, là kiến trúc hoàn toàn dựa trên MLP, đã cạnh tranh với các mô hình CNN và Transformer trên các nhiệm vụ xử lý hình ảnh. MLP-Mixer sử dụng

các lớp MLP cho cả không gian và kênh, đánh dấu sự quay trở lại đáng chú ý của MLP trong thị giác máy tính.

- **2022: Vision GNN (ViG)** được giới thiệu vào năm 2022, sử dụng GNN để xử lý hình ảnh trực tiếp, chứng minh tiềm năng của cấu trúc đồ thị trong thị giác máy tính
- **DHSL (Deep Hypergraph Structure Learning)** được giới thiệu, đây là một phương pháp học sâu để học cấu trúc của siêu đồ thị và áp dụng nó trong các nhiệm vụ như phân loại siêu đồ thị và học cấu trúc siêu đồ thị.
- **2023: ViHGNN (Vision HyperGraph Neural Network)** được đề xuất như một bước phát triển của ViG, sử dụng HGNN để biểu diễn hình ảnh dưới dạng siêu đồ thị động, cho phép nắm bắt các mối quan hệ phức tạp hơn giữa các phần tử hình ảnh.

## 5.4 Công Trình Dựa Trên

Các tác giả đã phát triển bài báo của họ dựa trên công trình trước đó về mạng Vision Graph Neural Networks (ViG). ViG là một phương pháp sử dụng mạng Graph Neural Networks (GNN) cho các tác vụ thị giác bằng cách chia hình ảnh thành các phân đoạn được coi là các nút trong đồ thị. Các tác giả của bài báo nhận ra tiềm năng của việc biểu diễn hình ảnh dưới dạng đồ thị bằng cách sử dụng ViG nhưng cũng xác định được những hạn chế của nó.

Bài báo mở rộng ý tưởng về ViG bằng cách đề xuất mạng Vision HyperGraph Neural Network (ViHGNN), vượt qua những hạn chế của ViG. Thay vì dựa vào biểu diễn đồ thị đơn giản với các cạnh đôi một, ViHGNN sử dụng siêu đồ thị để nắm bắt các mối quan hệ phức tạp hơn giữa các phân đoạn hình ảnh. Các tác giả lập luận rằng siêu đồ thị phù hợp hơn để biểu diễn hình ảnh vì chúng có thể kết nối nhiều nút (phân đoạn hình ảnh) trong một siêu cạnh duy nhất, cho phép biểu diễn chính xác hơn các mối quan hệ phức tạp.

Tóm lại, bài báo không phải là một hướng hoàn toàn mới mà là một sự phát triển dựa trên công trình trước đó về ViG, giải quyết những hạn chế của nó bằng cách sử dụng siêu đồ thị để biểu diễn hình

ảnh.

# 6 Methodology

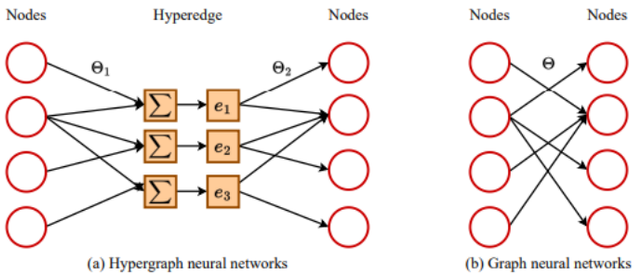
## 6.1 Hypergraph Neural Network (HNN)

Hypergraph Neural Network (HNN) là một mạng nơ-ron sử dụng hypergraph để mô hình hóa các mối quan hệ phức tạp giữa các nút. Một lớp convolutional tổng quát trong HNN được định nghĩa như sau:

$$X^{(l+1)} = D^{-1/2} H \sigma \left( W B^{-1} H^T \sigma \left( D^{-1/2} X^{(l)} \Theta_1 \right) \Theta_2 \right),$$

trong đó $W$ là ma trận trọng số hyperedge, $\Theta_1$ và $\Theta_2$ là các tham số có thể học được của lớp HNN, $\sigma$ là hàm kích hoạt, $X^{(l)}$ và $X^{(l+1)}$ lần lượt là các biểu diễn đầu vào và đầu ra của nút.

Cơ chế convolution của hypergraph có thể được hiểu như một quá trình truyền thông điệp hai giai đoạn, luân chuyển thông tin theo hướng "nút-hyperedge-nút". Việc nhân với $H^T$ đạt được sự tích hợp thông tin từ các nút vào các hyperedge, và nhân với $H$ giúp tích hợp thông tin ngược lại từ các hyperedge về các nút. Ma trận $D$ và $B$ thực hiện việc chuẩn hóa.




Hình 1: So sánh cơ chế truyền thông điệp trong (a) HGNN và (b) GNN.

GNN (Graph Neural Networks) và HGNN (Hypergraph Neural Networks) khác nhau ở ba khía cạnh chính:

- **Cấu trúc kết nối**: GNN sử dụng các cạnh đơn giản để kết nối trực tiếp giữa hai nút, giúp truyền thông tin chỉ giữa các cặp nút liền kề. Trong khi đó, HGNN sử dụng siêu đồ thị với các siêu

cạnh có thể kết nối nhiều nút cùng lúc, cho phép truyền thông tin phức tạp hơn.

- **Cơ chế truyền thông tin**: GNN thực hiện việc truyền thông tin trực tiếp giữa các nút qua các cạnh, giúp quá trình này đơn giản. Ngược lại, HGNN cho phép truyền thông tin hai chiều giữa các nút và siêu cạnh, tạo ra khả năng tổng hợp và phân phối thông tin từ nhiều nút, giúp tổng hợp thông tin tốt hơn.
- **Khả năng biểu diễn mối quan hệ phức tạp**: GNN có thể biểu diễn tốt các mối quan hệ đơn giản nhưng gặp hạn chế khi phải xử lý các mối quan hệ phức tạp. Trong khi đó, HGNN có khả năng mở rộng dễ dàng để biểu diễn các cấu trúc phức tạp, phù hợp với các bài toán đòi hỏi sự tổng hợp thông tin từ nhiều nguồn.

## 6.2 Input và Output

**Input:**

- Đầu vào của ViHGNN là một hình ảnh, được chia nhỏ thành các mảnh hình ảnh nhỏ (patches).
- Mỗi mảnh hình ảnh này sẽ được biểu diễn dưới dạng các vector đặc trưng thông qua các phương pháp trích xuất đặc trưng hình ảnh (như CNN hoặc các phương pháp dựa trên biến đổi Fourier).

**Output:**

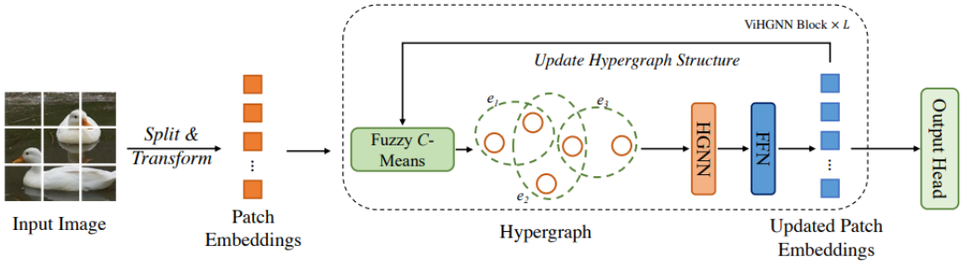
- Đầu ra của ViHGNN là các đặc trưng của hình ảnh sau khi đã qua quá trình xử lý thông qua các lớp convolutional của hypergraph.
- Những đặc trưng này có thể được sử dụng cho nhiều mục tiêu khác nhau như phân loại hình ảnh, nhận diện đối tượng hoặc các tác vụ khác liên quan đến thị giác máy tính.
- Cuối cùng, mô hình cung cấp đầu ra dự đoán tương ứng, ví dụ như nhãn phân loại hình ảnh hoặc tọa độ vị trí đối tượng trong bài toán localization.

## 6.3 Tổng quan về ViHGNN Framework

Framework của ViHGNN bao gồm hai cải tiến chính so với ViG gốc:

a. Đại diện cấu trúc hypergraph của một hình ảnh

b. Học cấu trúc hypergraph




Hình 2: Tổng quan về mô hình ViHNN.

ViHGNN Framework được cải tiến so với ViG gốc bằng cách sử dụng cấu trúc hypergraph để đại diện và học cấu trúc của hình ảnh. Trong mỗi lần lặp, hình ảnh được chia thành các mảnh và chuyển đổi thành các vector đặc trưng, sau đó các mảnh này được phân cụm bằng phương pháp Fuzzy C-Means để xây dựng hypergraph $G(I)$. Các lớp convolutional của hypergraph sau đó trao đổi thông tin giữa các nút qua cơ chế truyền thông điệp hai giai đoạn "nút-hyperedge-nút". Cuối cùng, áp dụng feed-forward network (FFN) để chiếu các đặc trưng nút vào cùng một miền và tăng cường sự đa dạng đặc trưng..

Framework ViHNN lặp lại nhiệm vụ này bằng cách liên tục cập nhật hypergraph và đánh giá đại diện đặc trưng của hypergraph.

## 6.4 Cấu trúc Hpergraph của hình ảnh

Hypergraph được xây dựng bằng cách coi hình ảnh là một tập hợp các nút không có thứ tự và các đặc trưng liên kết. Phương pháp Fuzzy C-Means được sử dụng để tạo ra các cụm nút đại diện cho các tập hợp mảnh hình ảnh, tương tự như hyperedge. Cấu trúc này cho phép các nút nằm trong nhiều

hyperedge khác nhau, giúp mô hình hóa các mối quan hệ phức tạp và đa dạng giữa các mảnh hình ảnh.

### 6.5 Tối ưu hóa cấu trúc hypergraph thích ứng và tối ưu hóa end-to-end

Quá trình tối ưu hóa cấu trúc hypergraph thích ứng nhằm loại bỏ hoặc giảm các kết nối không liên quan đến nhiệm vụ, đồng thời tăng cường các kết nối quan trọng. Mục tiêu là đạt được một đại diện hoàn chỉnh và chính xác của các mối quan hệ bậc cao trong các mảnh hình ảnh. Để thực hiện điều này, một vòng lặp phản hồi ("feedback loop") được thiết lập, trong đó các embeddings của mảnh hình ảnh và hypergraph hỗ trợ lẫn nhau, giúp cải thiện đồng thời cả hai. Bên cạnh đó, việc giới hạn số lượng hyperedge giúp ngăn chặn các cấu trúc không cần thiết, tăng cường hiệu quả của hypergraph.

ViHGNN thực hiện quá trình tối ưu hóa toàn diện ("end-to-end optimization"), trong đó các cấu trúc hypergraph được cập nhật thích ứng dựa trên các embeddings của mảnh hình ảnh học được. Quá trình này hoạt động độc lập với việc tính toán gradient và không ảnh hưởng đến quá trình backpropagation, nhưng vẫn đảm bảo đạt được các mục tiêu cụ thể của nhiệm vụ, chẳng hạn như cross-entropy loss cho phân loại hình ảnh và localization loss cho nhận diện đối tượng.

### 6.6 Chi tiết cấu hình của ViHGNN

Bảng 3 và Bảng 4 trình bày các cấu hình chi tiết của series ViHGNN Isotropic và ViHGNN Pyramid, trong đó $D$ là kích thước đặc trưng, $h$ là tỉ lệ kích thước ẩn trong FFN, $E$ là số lượng hyperedge trong HNN, và $H \times W$ là kích thước đầu vào của hình ảnh. Hai bảng này cung cấp cái nhìn tổng quan về các thông số và hiệu năng của các mô hình ViHGNN, cho phép lựa chọn cấu hình phù hợp với các yêu cầu cụ thể của bài toán.

Table 1. Detailed settings of Isotropic ViHGNN series. $D$: feature dimension, $h$: hidden dimension ratio in FFN, $E$: number of hyperedges in HGNN, $H \times W$: input image size. 'Ti' denotes tiny, 'S' denotes small, 'M' denotes medium, and 'B' denotes base.

| Model | Depth | Dimension $D$ | Hyperedges ($E$) | Parameters (M) | FLOPs (B) |
|---|---|---|---|---|---|
| Isotropic ViHGNN-Ti | 12 | 192 | 50 | 8.2 | 1.8 |
| Isotropic ViHGNN-S | 16 | 320 | 50 | 23.2 | 5.6 |
| Isotropic ViHGNN-B | 16 | 640 | 50 | 88.1 | 19.4 |

Hình 3: Mô tả cấu hình chi tiết của series ViHGNN Isotropic.

- Mô hình Isotropic ViHGNN gồm ba phiên bản: Ti (Tiny), S (Small), và B (Base). Sự khác biệt chính nằm ở độ sâu mô hình, kích thước đặc trưng, và số lượng hyperedge.
- **FLOPs** là thước đo hiệu suất tính toán, biểu thị số lượng phép toán dấu phẩy động mà mô hình có thể thực hiện trong một giây.
- **ViHGNN-B** có số lượng tham số và FLOPs lớn nhất, phù hợp cho các ứng dụng yêu cầu học đặc trưng phức tạp nhưng đòi hỏi nhiều tài nguyên tính toán. **ViHGNN-S** cân bằng giữa hiệu suất và tài nguyên, phù hợp cho các ứng dụng có yêu cầu trung bình. **ViHGNN-Ti** có số lượng tham số và FLOPs ít nhất, thích hợp cho các ứng dụng cần tiết kiệm tài nguyên.

Table 2. Detailed settings of Pyramid ViHGNN series. $D$: feature dimension, $h$: hidden dimension ratio in FFN, $E$: number of hyperedges in HGNN, $H \times W$: input image size. 'Ti' denotes tiny, 'S' denotes small, 'M' denotes medium, and 'B' denotes base.

| Stage | Output size | Pyramid ViHGNN-Ti | Pyramid ViHGNN-S | Pyramid ViHGNN-M | Pyramid ViHGNN-B |
|---|---|---|---|---|---|
| Stem | $\frac{H}{4} \times \frac{W}{4}$ | Conv×3 | Conv×3 | Conv×3 | Conv×3 |
| Stage 1 | $\frac{H}{4} \times \frac{W}{4}$ | $\begin{bmatrix} D=48 \\ h=4 \\ E=50 \end{bmatrix} \times 2$ | $\begin{bmatrix} D=80 \\ h=4 \\ E=50 \end{bmatrix} \times 2$ | $\begin{bmatrix} D=96 \\ h=4 \\ E=50 \end{bmatrix} \times 2$ | $\begin{bmatrix} D=128 \\ h=4 \\ E=50 \end{bmatrix} \times 2$ |
| Downsample | $\frac{H}{8} \times \frac{W}{8}$ | Conv | Conv | Conv | Conv |
| Stage 2 | $\frac{H}{8} \times \frac{W}{8}$ | $\begin{bmatrix} D=96 \\ h=4 \\ E=50 \end{bmatrix} \times 2$ | $\begin{bmatrix} D=160 \\ h=4 \\ E=50 \end{bmatrix} \times 2$ | $\begin{bmatrix} D=192 \\ h=4 \\ E=50 \end{bmatrix} \times 2$ | $\begin{bmatrix} D=256 \\ h=4 \\ E=50 \end{bmatrix} \times 2$ |
| Downsample | $\frac{H}{16} \times \frac{W}{16}$ | Conv | Conv | Conv | Conv |
| Stage 3 | $\frac{H}{16} \times \frac{W}{16}$ | $\begin{bmatrix} D=240 \\ h=4 \\ E=50 \end{bmatrix} \times 6$ | $\begin{bmatrix} D=400 \\ h=4 \\ E=50 \end{bmatrix} \times 6$ | $\begin{bmatrix} D=384 \\ h=4 \\ E=50 \end{bmatrix} \times 16$ | $\begin{bmatrix} D=512 \\ h=4 \\ E=50 \end{bmatrix} \times 18$ |
| Downsample | $\frac{H}{32} \times \frac{W}{32}$ | Conv | Conv | Conv | Conv |
| Stage 4 | $\frac{H}{32} \times \frac{W}{32}$ | $\begin{bmatrix} D=384 \\ h=4 \\ E=50 \end{bmatrix} \times 2$ | $\begin{bmatrix} D=640 \\ h=4 \\ E=50 \end{bmatrix} \times 2$ | $\begin{bmatrix} D=768 \\ h=4 \\ E=50 \end{bmatrix} \times 2$ | $\begin{bmatrix} D=1024 \\ h=4 \\ E=50 \end{bmatrix} \times 2$ |
| Head | $1 \times 1$ | Pooling & MLP | Pooling & MLP | Pooling & MLP | Pooling & MLP |
| Parameters (M) | | 12.3 | 28.5 | 52.4 | 94.4 |
| FLOPs (B) | | 2.3 | 6.3 | 10.7 | 18.1 |

Hình 4: Mô tả cấu hình chi tiết của series ViHGNN Pyramid.

- Mô hình Pyramid ViHGNN được thiết kế với cấu trúc tầng (Stage) từ Stem đến Stage 4, trong đó kích thước đặc trưng (D) và tỷ lệ ẩn (h) tăng dần qua từng tầng.
- **Pyramid ViHGNN-B** có cấu trúc phức tạp nhất với các tầng có kích thước đặc trưng lớn, giúp mô hình này nắm bắt được các đặc trưng chi tiết hơn trong hình ảnh.
- **Pyramid ViHGNN-B** yêu cầu nhiều tài nguyên tính toán nhất với số lượng tham số và FLOPs lớn nhất. Các mô hình nhỏ hơn như **Pyramid ViHGNN-Ti** và **Pyramid ViHGNN-S** có cấu trúc đơn giản hơn, phù hợp cho các ứng dụng đòi hỏi ít tài nguyên hơn nhưng vẫn đảm bảo khả năng học các đặc trưng quan trọng.

**Nhận xét thực tiễn:**

- Các cấu hình được trình bày trong Bảng 3 và Bảng 4 cung cấp sự linh hoạt trong việc lựa chọn mô hình dựa trên yêu cầu cụ thể của từng ứng dụng thực tiễn. Đối với các tác vụ đòi hỏi độ chính xác cao nhưng giới hạn về tài nguyên tính toán, các mô hình nhỏ hơn như ViHGNN-Ti hoặc ViHGNN-S có thể là lựa chọn phù hợp nhờ vào số lượng tham số và FLOPs thấp hơn. Trong khi đó, các mô hình lớn hơn như ViHGNN-B sẽ phù hợp với các ứng dụng yêu cầu khả năng xử lý mạnh mẽ, chẳng hạn như phân loại hình ảnh hoặc nhận diện đối tượng trong các hệ thống thị giác máy tính phức tạp.
- Việc thiết kế các mô hình theo cấu trúc Pyramid cho phép xử lý hiệu quả hơn khi kích thước đầu ra giảm dần qua các giai đoạn, điều này rất quan trọng trong việc giảm thiểu chi phí tính toán đồng thời giữ được tính chính xác của mô hình. Điều này mang lại lợi ích đặc biệt trong các ứng dụng thời gian thực, nơi việc tối ưu hóa hiệu năng và tài nguyên là yếu tố quyết định sự thành công của hệ thống.
- Tóm lại, việc lựa chọn mô hình cần dựa trên sự cân nhắc kỹ lưỡng giữa yêu cầu thực tiễn về độ chính xác và tài nguyên tính toán sẵn có, nhằm đảm bảo hiệu quả cao nhất trong các ứng dụng cụ thể.

## 6.7 Lợi ích và Bất lợi

### Lợi ích

- Biểu diễn mối quan hệ phức tạp: ViHGNN có khả năng biểu diễn các mối quan hệ phức tạp và bậc cao hơn nhờ việc sử dụng cấu trúc hypergraph. Điều này giúp mô hình nắm bắt thông tin tốt hơn so với các mô hình truyền thống dựa trên graph.
- Truyền thông tin hai chiều: Cơ chế truyền thông điệp "nút-hyperedge-nút"cho phép tích hợp thông tin giữa các nút và các siêu cạnh, mang lại khả năng tổng hợp và truyền đạt thông tin tốt hơn giữa các mảnh hình ảnh.
- Khả năng thích ứng cao: Quá trình tối ưu hóa hypergraph giúp loại bỏ các kết nối không cần thiết và tăng cường những kết nối quan trọng, dẫn đến mô hình học được đại diện chính xác và phù hợp hơn với dữ liệu.
- Hiệu suất cao với cấu trúc tầng: Các phiên bản Pyramid ViHGNN với cấu trúc tầng (stage) cho phép tối ưu hóa việc xử lý các đặc trưng hình ảnh qua nhiều tầng, giảm thiểu chi phí tính toán trong khi vẫn giữ được khả năng nắm bắt chi tiết các đặc trưng quan trọng.

### Bất lợi:

- Tài nguyên tính toán lớn: Các mô hình lớn như ViHGNN-B có yêu cầu về tài nguyên tính toán cao (FLOPs và tham số lớn), điều này có thể gây ra khó khăn trong việc triển khai trên các hệ thống có tài nguyên hạn chế.
- Phức tạp trong việc thiết kế và tối ưu hóa: Việc thiết kế và tối ưu hóa hypergraph có thể trở nên phức tạp và khó khăn, đặc biệt khi cấu trúc hypergraph không dễ định nghĩa hoặc cần phải tinh chỉnh nhiều tham số khác nhau để đạt được hiệu năng tốt nhất.
- Khả năng mở rộng hạn chế: Mặc dù ViHGNN mạnh mẽ trong việc biểu diễn các mối quan hệ phức tạp, nhưng khả năng mở rộng đối với các bài toán lớn hoặc những bộ dữ liệu quá đa dạng có thể gặp hạn chế, đặc biệt khi số lượng hyperedge hoặc nút trong hypergraph tăng mạnh.

# 7 Experimental and Evaluations

## 7.1 Quy Trình Thực Hiện

Bài báo thực hiện quy trình thí nghiệm qua 3 phần chính, nhằm đánh giá hiệu suất của mô hình ViHGNN trên các tác vụ thị giác máy tính:

- **Áp dụng ViHGNN trên nhiệm vụ phân loại ảnh:** Thựuc nghiệm áp dụng ViHGNN trên một trong những tác vụ cơ bản nhất trong thị giác máy tính, giúp kiểm tra khả năng của mô hình trong việc phân loại chính xác các hình ảnh.
- **Thử nghiệm để đánh giá ảnh hưởng của từng thành phần trong mô hình ViHGNN:** Để hiểu rõ hơn về tầm quan trọng của từng thành phần trong ViHGNN, các tác giả đã thực hiện các thí nghiệm ablation study nhằm loại bỏ hoặc thay đổi một số thành phần nhất định.
- **Áp dụng trên nhiệm vụ phát hiện đối tượng:** Sau khi đánh giá mô hình trên tác vụ phân loại, mô hình được áp dụng trên một tác vụ phức tạp hơn là phát hiện đối tượng, giúp đánh giá khả năng tổng quát hóa của ViHGNN.

#### Các thiết lập cho quy trình thử nghiệm (Experimental Settings)

Trong phần này, thông tin về các tập dữ liệu được sử dụng, các mô hình baseline được so sánh, và cài đặt siêu tham số cho các thí nghiệm sẽ được trình bày.

- **Datasets (Tập dữ liệu):** Các thử nghiệm sử dụng hai tập dữ liệu phổ biến là:
  - **Đối với nhiệm vụ phân loại ảnh:** sử dụng tập dữ liệu **ImageNet ILSVRC 2012**, bao gồm 120 triệu hình ảnh huấn luyện và 50.000 hình ảnh xác thực thuộc 1000 danh mục.
  - **Đối với nhiệm vụ phát hiện đối tượng:** sử dụng tập dữ liệu **COCO 2017** với 80 danh mục đối tượng, bao gồm 118.000 hình ảnh huấn luyện và 5.000 hình ảnh xác thực.
- **Baselines (Mô hình baseline):** Để đảm bảo sự so sánh công bằng, các mô hình baseline được lựa chọn dựa trên nghiên cứu ViG ban đầu. Hai loại kiến trúc mạng được sử dụng để so sánh là

đẳng hướng (Isotropic) và kim tự tháp (Pyramid).

  - **Isotropic:** Duy trì kích thước đặc trưng xuyên suốt mạng, phù hợp cho việc mở rộng quy mô và tăng tốc phần cứng. Isotropic ViHGNN sẽ được so sánh với các mô hình đẳng hướng khác như ResMLP, ConvMixer, ViT, DeiT.
  - **Pyramid:** Giảm dần kích thước không gian của bản đồ đặc trưng khi mạng sâu hơn, tận dụng tính chất bất biến tỷ lệ của ảnh để tạo ra các đặc trưng đa tỷ lệ. Pyramid ViHGNN sẽ được so sánh với các mô hình như ResNet, BoTNet, PVT, CVT, Swin Transformer, CycleMLP, Poolformer.

- **Hyper-parameters Settings (Cài đặt siêu tham số):** Các mô hình ViHGNN được huấn luyện sử dụng các chiến lược huấn luyện phổ biến trong DeiT cho nhiệm vụ phân loại ảnh trên ImageNet. Đối với nhiệm vụ phát hiện đối tượng trên COCO, sử dụng RetinaNet và Mask R-CNN làm framework chung, huấn luyện trên lịch trình "1x"và tính toán FLOPs với kích thước đầu vào $1280 \times 800$. Chi tiết phần cài đặt siêu tham số được sử dụng để huấn luyện các mô hình ViHGNN trên tập dữ liệu ImageNet cho nhiệm vụ phân loại hình ảnh là như sau (Cụ thể, giá trị của các siêu tham số cho các mô hình ViHGNN với kích thước khác nhau (Tiny, Small, Medium, Base):
  - **Epochs:** tất cả các mô hình ViHGNN đều được huấn luyện trong 300 epoch.
  - **Optimizer:** AdamW được sử dụng làm trình tối ưu hóa cho tất cả các mô hình.
  - **Batch size:** 1024 cho tất cả mô hình ViHGNN
  - **Start learning rate (LR):** tốc độ học sẽ giảm dần theo hàm cosine.
  - **Warmup epochs:** có 20 epoch khởi động.
  - **Weight decay:** 0.05
  - **Label smoothing:** 0.1
  - **Stochastic path:** 0.1, 0.1, 0.1 và 0.3 tương ứng lần lượt cho các mô hình ViHGNN-Ti,

ViHGNN-S, ViHGNN-M và ViHGNN-B

  - **Repeated augment:** kỹ thuật này được sử dụng cho tất cả các mô hình.
  - **RandAugment:** kỹ thuật này được sử dụng cho tất cả các mô hình.
  - **Mixup prob:** xác suất sử dụng Mixup là 80
  - **Cutmix prob:** xác suất sử dụng Cutmix là 100
  - **Random erasing prob:** xác suất sử dụng xóa ngẫu nhiên là 25
  - **Exponential moving average:** 0.99996

- **Nền tảng:** Thực hiện bằng PyTorch và huấn luyện trên 8 GPU NVIDIA V100 của một phiên bản AWS EC2.

## 7.2 Kết Quả

**1. Áp dụng ViHGNN trên nhiệm vụ phân loại ảnh**

ViHGNN được áp dụng để so sánh và đánh giá trên nhiệm vụ phân loại ảnh với hai kiến trúc khác nhau:

- **Kiến trúc Isotropic:** Kết quả sau khi chạy thực nghiệm để so sánh hiệu suất của ViHGNN với các mô hình isotropic khác được thể hiện trong hình sau:

Table 4. Results of ViHGNN and other isotropic networks on ImageNet. ♠ CNN, ♥ Transformer, ♣ MLP, ♦ GNN, ■ HGNN.

| Model | Resolution | Params (M) | FLOPs (B) | Top-1 | Top-5 |
|---|---|---|---|---|---|
| ♠ ResMLP-S12 conv3x3 [55] | 224×224 | 16.7 | 3.2 | 77.0 | - |
| ♠ ConvMixer-768/32 [57] | 224×224 | 21.1 | 20.9 | 80.2 | - |
| ♠ ConvMixer-1536/20 [57] | 224×224 | 51.6 | 51.4 | 81.4 | - |
| ♥ ViT-B/16 [11] | 384×384 | 86.4 | 55.5 | 77.9 | - |
| ♥ DeiT-Ti [56] | 224×224 | 5.7 | 1.3 | 72.2 | 91.1 |
| ♥ DeiT-S [56] | 224×224 | 22.1 | 4.6 | 79.8 | 95.0 |
| ♥ DeiT-B [56] | 224×224 | 86.4 | 17.6 | 81.8 | 95.7 |
| ♣ ResMLP-S24 [55] | 224×224 | 30 | 6.0 | 79.4 | 94.5 |
| ♣ ResMLP-B24 [55] | 224×224 | 116 | 23.0 | 81.0 | 95.0 |
| ♣ Mixer-B/16 [54] | 224×224 | 59 | 11.7 | 76.4 | - |
| ♦ Isotropic ViG-Ti | 224×224 | 7.1 | 1.3 | 73.9 | 92.0 |
| ♦ Isotropic ViG-S | 224×224 | 22.7 | 4.5 | 80.4 | 95.2 |
| ♦ Isotropic ViG-B | 224×224 | 86.8 | 17.7 | 82.3 | 95.9 |
| ■ ViHGNN-Ti (ours) | 224×224 | 8.2 | 1.8 | **74.3** | **92.5** |
| ■ ViHGNN-S (ours) | 224×224 | 23.2 | 5.6 | **81.5** | **95.7** |
| ■ ViHGNN-B (ours) | 224×224 | 88.1 | 19.4 | **82.9** | **96.2** |

Hình 5: Kết quả so sánh hiệu suất của ViHGNN với các mô hình isotropic khác trên nhiệm vụ phân loại ảnh

Cụ thể hơn, bảng trong hình 5 này so sánh hiệu suất của mô hình ViHGNN được đề xuất với các mô hình mạng nơ-ron khác về độ chính xác phân loại top-1 và top-5, số lượng tham số (Params) và hoạt động dấu chấm động (FLOPs). Mỗi loại mô hình được ký hiệu bằng một biểu tượng riêng biệt như trong hình đã mô tả và có tổng 5 loại mô hình được so sánh là CNN, Transformer, MLP, GNN và HGNN. Từ hình 5 trên, ta có thể đưa ra các kết luận sau:

– **Hiệu suất:** Nhìn chung, ViHGNN vượt trội hơn các mạng đẳng hướng khác về độ chính xác. Ví dụ, ViHGNN-S đạt độ chính xác top-1 là 81,5%, vượt trội hơn DeiT-Ti 1,1% trong khi vẫn duy trì chi phí tính toán tương đương. Điều này cho thấy hiệu quả của việc sử dụng siêu đồ thị để biểu diễn hình ảnh và mạng nơ-ron siêu đồ thị cho các tác vụ thị giác.

  - **Chi phí tính toán:** Hình 5 cũng cho thấy ViHGNN có chi phí tính toán (Params và FLOPs) tương đương với các mạng đẳng hướng khác. Điều này cho thấy ViHGNN có thể đạt được hiệu suất cao hơn mà không cần tăng chi phí tính toán.
  - **So sánh với ViG:** Hình 5 cũng so sánh ViHGNN với ViG, một phương pháp sử dụng biểu diễn đồ thị cho hình ảnh. Kết quả cho thấy ViHGNN luôn vượt trội hơn ViG với cùng kích thước mô hình và chi phí tính toán. Điều này cho thấy việc sử dụng siêu đồ thị để biểu diễn hình ảnh mang lại hiệu quả tốt hơn so với đồ thị thông thường.

- **Kiến trúc Pyramid:** Kết quả sau khi chạy thực nghiệm để so sánh hiệu suất của ViHGNN với các mô hình pyramid khác được thể hiện trong hình sau:

Table 5. Results of Pyramid ViHGNN and other pyramid networks on ImageNet. ♠ CNN, ♥ Transformer, ♣ MLP, ♦ GNN, ■ HGNN.

| Model | Resolution | Params (M) | FLOPs (B) | Top-1 | Top-5 |
|---|---|---|---|---|---|
| ♠ ResNet-18 [22, 65] | 224×224 | 12 | 1.8 | 70.6 | 89.7 |
| ♠ ResNet-50 [22, 65] | 224×224 | 25.6 | 4.1 | 79.8 | 95.0 |
| ♠ ResNet-152 [22, 65] | 224×224 | 60.2 | 11.5 | 81.8 | 95.9 |
| ♠ BoTNet-T3 [51] | 224×224 | 33.5 | 7.3 | 81.7 | - |
| ♠ BoTNet-T3 [51] | 224×224 | 54.7 | 10.9 | 82.8 | - |
| ♠ BoTNet-T3 [51] | 256×256 | 75.1 | 19.3 | 83.5 | - |
| ♥ PVT-Tiny [62] | 224×224 | 13.2 | 1.9 | 75.1 | - |
| ♥ PVT-Small [62] | 224×224 | 24.5 | 3.8 | 79.8 | - |
| ♥ PVT-Medium [62] | 224×224 | 44.2 | 6.7 | 81.2 | - |
| ♥ PVT-Large [62] | 224×224 | 61.4 | 9.8 | 81.7 | - |
| ♥ CvT-13 [66] | 224×224 | 20 | 4.5 | 81.6 | - |
| ♥ CvT-21 [66] | 224×224 | 32 | 7.1 | 82.5 | - |
| ♥ CvT-21 [66] | 384×384 | 32 | 24.9 | 83.3 | - |
| ♥ Swin-T [41] | 224×224 | 29 | 4.5 | 81.3 | 95.5 |
| ♥ Swin-S [41] | 224×224 | 50 | 8.7 | 83.0 | 96.2 |
| ♥ Swin-B [41] | 224×224 | 88 | 15.4 | 83.5 | 96.5 |
| ♣ CycleMLP-B2 [5] | 224×224 | 27 | 3.9 | 81.6 | - |
| ♣ CycleMLP-B3 [5] | 224×224 | 38 | 6.9 | 82.4 | - |
| ♣ CycleMLP-B4 [5] | 224×224 | 52 | 10.1 | 83.0 | - |
| ♣ Poolformer-S12 [74] | 224×224 | 12 | 2.0 | 77.2 | 93.5 |
| ♣ Poolformer-S36 [74] | 224×224 | 31 | 5.2 | 81.4 | 95.5 |
| ♣ Poolformer-M48 [74] | 224×224 | 73 | 11.9 | 82.5 | 96.0 |
| ♦ Pyramid ViG-Ti [18] | 224×224 | 10.7 | 1.7 | 78.2 | 94.2 |
| ♦ Pyramid ViG-S [18] | 224×224 | 27.3 | 4.6 | 82.1 | 96.0 |
| ♦ Pyramid ViG-M [18] | 224×224 | 51.7 | 8.9 | 83.1 | 96.4 |
| ♦ Pyramid ViG-B [18] | 224×224 | 92.6 | 16.8 | 83.7 | 96.5 |
| ■ Pyramid ViHGNN-Ti (ours) | 224×224 | 12.3 | 2.3 | **78.9** | **94.6** |
| ■ Pyramid ViHGNN-S (ours) | 224×224 | 28.5 | 6.3 | **82.5** | **96.3** |
| ■ Pyramid ViHGNN-M (ours) | 224×224 | 52.4 | 10.7 | **83.4** | **96.5** |
| ■ Pyramid ViHGNN-B (ours) | 224×224 | 94.4 | 18.1 | **83.9** | **96.7** |

Hình 6: Kết quả so sánh của mô hình Pyramid ViHGNN được đề xuất với các mạng Pyramid khác trên ImageNet.

Chi tiết hơn thì bảng trong hình 6 này so sánh hiệu suất của mô hình ViHGNN được đề xuất với các mô hình mạng nơ-ron khác về độ phân giải của hình ảnh, độ chính xác phân loại top-1 và top-5, số lượng tham số (Params) và hoạt động dấu chấm động (FLOPs). Mỗi loại mô hình

được ký hiệu bằng một biểu tượng riêng biệt như trong hình đã mô tả và có tổng 5 loại mô hình được so sánh là CNN, Transformer, MLP, GNN và HGNN. Từ hình 6 trên, ta có thể đưa ra các kết luận sau:

- **Hiệu suất:** Nhìn chung, các mô hình Pyramid ViHGNN (Ti, S, M, B) đạt hiệu suất tương đương hoặc vượt trội hơn so với các mô hình hàng đầu khác trong cùng nhóm kiến trúc. Đặc biệt, Pyramid ViHGNN-B đạt độ chính xác top-1 là 83.9%, vượt qua các mô hình mạnh như BoTNet-T3, CvT-21 và Swin-B. Điều này cho thấy khả năng nắm bắt các mối quan hệ phức tạp giữa các phần tử trong ảnh của mô hình ViHGNN, nhờ vào việc sử dụng siêu đồ thị để biểu diễn ảnh.
- **So sánh với ViG:** Pyramid ViHGNN thường cho kết quả tốt hơn so với Pyramid ViG trên cùng cấu hình. Ví dụ, Pyramid ViHGNN-S đạt độ chính xác top-1 là 82.5%, trong khi Pyramid ViG-S chỉ đạt 82.1%. Sự khác biệt này cho thấy việc sử dụng siêu đồ thị trong ViHGNN giúp cải thiện khả năng biểu diễn ảnh so với việc sử dụng đồ thị thông thường trong ViG.

#### 2. Thử nghiệm để đánh giá ảnh hưởng của từng thành phần trong mô hình ViHGNN

Tác giả bài báo tiếp tục thực hiện một số thử nghiệm loại bỏ để đánh giá tác động của các thành phần khác nhau trong mô hình ViHGNN trên hiệu suất phân loại ảnh (sử dụng Isotropic ViHGNN-Ti làm kiến trúc cơ sở). Các yếu tố được phân tích bao gồm:

a. **Phương pháp xây dựng siêu đồ thị:**

So sánh hiệu quả của k-NN, K-Means và Fuzzy C-Means trong việc xây dựng siêu đồ thị. Kết quả so sánh được trình bày trong bảng ở hình sau:

Table 6. Ablation study on ImageNet for classification task.

| Type of $G(\boldsymbol{I})$ | Num. of $E$ | HSL module | Params (M) | FLOPs (B) | Top-1 | Top-5 |
|---|---|---|---|---|---|---|
| $k$-NN | 25 | ✗ | 7.9 | 1.5 | 72.4 | 89.7 |
| $k$-NN | 25 | ✓ | 8.2 | 1.9 | 72.8 | 90.2 |
| $k$-NN | 50 | ✗ | 8.9 | 2.4 | 72.5 | 89.6 |
| $k$-NN | 50 | ✓ | 9.4 | 2.9 | 73.0 | 90.1 |
| $K$-Means | 25 | ✗ | 7.3 | 1.4 | 72.6 | 90.1 |
| $K$-Means | 25 | ✓ | 8.0 | 2.1 | 73.1 | 90.3 |
| $K$-Means | 50 | ✗ | 8.4 | 2.5 | 73.5 | 91.1 |
| $K$-Means | 50 | ✓ | 9.2 | 2.8 | 73.7 | 92.3 |
| Fuzzy $C$-Means | 25 | ✗ | 8.2 | 2.2 | 72.9 | 90.4 |
| Fuzzy $C$-Means | 25 | ✓ | 8.8 | 2.9 | 73.5 | 90.9 |
| Fuzzy $C$-Means | 50 | ✗ | 9.1 | 3.2 | 74.1 | 92.2 |
| Fuzzy $C$-Means | 50 | ✓ | 9.7 | 3.8 | 74.9 | 92.9 |

Hình 7: Kết quả so sánh đánh giá ảnh hưởng của các phương pháp xây dựng siêu đồ thị

Cụ thể, bảng trong hình 7 này so sánh hiệu suất của mô hình khi sử dụng ba phương pháp xây dựng siêu đồ thị khác nhau là k-NN, K-Means và Fuzzy C-Means, với số lượng siêu cạnh (Num. of E) là 25 và 50, và có sử dụng hoặc không sử dụng mô-đun học cấu trúc siêu đồ thị (HSL module). Kết quả nghiên cứu từ hình 7 cho thấy:

- **So sánh phương pháp xây dựng siêu đồ thị: :** Fuzzy C-Means cho kết quả tốt nhất trong 3 phương pháp, với độ chính xác Top-1 cao hơn và chi phí tính toán chỉ tăng nhẹ so với K-Means. Điều này cho thấy Fuzzy C-Means tạo ra siêu đồ thị phù hợp nhất để mô hình hóa các mối quan hệ phức tạp trong ảnh.
- **Tác dụng của module HSL:** Việc sử dụng mô-đun học cấu trúc siêu đồ thị (HSL) cũng giúp cải thiện hiệu suất của mô hình (khoảng 0.5-0.8% độ chính xác Top-1) với chi phí tính toán tăng không đáng kể. Kết quả này chứng minh hiệu quả của việc học cấu trúc siêu đồ thị trong mô hình ViHGNN.
- **Ảnh hưởng của số lượng siêu cạnh:** Số lượng siêu cạnh cũng ảnh hưởng đến hiệu suất của mô hình. Nói chung, việc tăng số lượng siêu cạnh có thể giúp cải thiện độ chính xác, nhưng

đồng thời cũng làm tăng chi phí tính toán.

b. **Hiệu suất của các phương pháp phân cụm khác nhau:**

Bên cạnh đó, tác giả cũng thực nghiệm để so sánh hiệu suất của mô hình ViHGNN khi sử dụng các phương pháp phân cụm khác nhau để cập nhật cấu trúc siêu đồ thị. Sau khi thí nghiệm thì kết quả được như hình sau:

Table 7. Comparison of various clustering methods with Istropic ViHGNN-Ti as the backbone model.

| Clustering Methods | DBSCAN | Mean Shift Clustering | Spectral Clustering | Fuzzy C-Means |
|---|---|---|---|---|
| Top-1 | 74.2 | 74.5 | 74.7 | **74.9** |
| Top-5 | 92.6 | 92.7 | 92.8 | **92.9** |

Hình 8: Kết quả so sánh của mô hình Isotropic ViHGNN-Ti khi sử dụng các phương pháp phân cụm khác nhau

Chi tiết hơn thì bảng trong hình 8 này so sánh độ chính xác top-1 và top-5 của mô hình Isotropic ViHGNN-Ti khi sử dụng các phương pháp phân cụm khác nhau để cập nhật cấu trúc siêu đồ thị. Các phương pháp được so sánh bao gồm DBSCAN, Mean Shift Clustering, Spectral Clustering và Fuzzy C-Means. Kết quả cho thấy:

- **So sánh hiệu suất:** Fuzzy C-Means cho kết quả tốt nhất trong số các phương pháp được so sánh, với độ chính xác Top-1 và Top-5 cao nhất (74.9% và 92.9%).
- Nhìn chung thì Fuzzy C-Means linh hoạt hơn và ít nhạy cảm với nhiễu hơn so với các phương pháp thay thế. Điều này cho phép Fuzzy C-Means tìm ra các siêu cạnh tốt hơn, phù hợp hơn với sự phân bố của các patch, từ đó nắm bắt hiệu quả hơn các mối quan hệ phức tạp giữa các patch nhúng và cuối cùng cung cấp biểu diễn chính xác hơn về hình ảnh, dẫn đến hiệu suất được cải thiện trong mô hình.

c. **Số vòng lặp cập nhật:**

Tác giả đánh giá sự ảnh hưởng của số lần cập nhật vòng lặp bằng cách thử nghiệm khi sử dụng 1, 2 và 3 vòng lặp cập nhật cho mô hình Isotropic ViHGNN-Ti. Kết quả được trình bày như sau:

Table 8. Comparison of various update loops with Istropic ViHGNN-Ti as the backbone model.

| # Update Loops | 1 | 2 | 3 |
|---|---|---|---|
| Top-1 | 74.9 | **75.0** | 74.4 |
| Top-5 | 92.9 | **93.0** | 92.7 |

Hình 9: Kết quả so sánh hiệu suất của mô hình Istropic ViHGNN-Ti với số lần lặp cập nhật khác nhau.

Theo kết quả được trình bày trong bảng ở hình 9, việc sử dụng 2 hoặc 3 vòng lặp cập nhật không mang lại sự khác biệt đáng kể về độ chính xác so với việc chỉ sử dụng 1 vòng lặp. Điều này cho thấy rằng đối với mô hình Isotropic ViHGNN-Ti, một vòng lặp cập nhật có thể đã đủ để đạt được hiệu suất tối ưu ở quy mô này. Việc sử dụng nhiều vòng lặp hơn chỉ làm tăng thêm số lượng tham số và FLOPs của mô hình mà không mang lại lợi ích đáng kể.

d. **Số lượng siêu cạnh:**

Tác giả phân tích ảnh hưởng của số lượng siêu cạnh (E), là số cụm được tạo ra bởi thuật toán Fuzzy C-Means bằng cách thử nghiệm sau đó đánh giá hiệu suất của mô hình Pyramid ViHGNN-Ti khi sử dụng số lượng siêu cạnh (E) khác nhau. Kết quả sau thực nghiệm như sau:

Table 9. Effects of *E* for Pyramid ViHGNN-Ti.

| E | 50 | 50 (↓) | 100 | 100 (↓) |
|---|---|---|---|---|
| Top-1 | 78.9 | 79.4 | 77.6 | 78.9 |
| Top-5 | 94.6 | 95.0 | 94.2 | 94.7 |

Hình 10: Kết quả của mô hình Pyramid ViHGNN-Ti khi sử dụng số lượng siêu cạnh (E) khác nhau.

Kết quả nhận được cho thấy:

- Việc tăng số lượng siêu cạnh (từ 50 lên 100) không phải lúc nào cũng dẫn đến hiệu suất tốt hơn. Đối với mô hình Pyramid ViHGNN-Ti, sử dụng E = 100 cho kết quả kém hơn so với E = 50.
- Việc giảm một nửa số lượng siêu cạnh ở mỗi tầng (50 (↓)) cho kết quả tốt hơn so với việc sử dụng số lượng siêu cạnh cố định (50) hoặc tăng gấp đôi (100).
- Nhìn chung, việc phân bổ số lượng siêu cạnh phù hợp cho từng tầng có thể mang lại hiệu quả tốt hơn. Mô hình với E = 50 (↓) đạt hiệu suất tốt nhất trong số các cấu hình được thử nghiệm, cho thấy rằng việc giảm số lượng siêu cạnh cho các tầng sau có thể giúp mô hình học hỏi hiệu quả hơn.
- Tóm lại, việc tăng số lượng siêu cạnh không đảm bảo hiệu suất tốt hơn và có thể dẫn đến chi phí tính toán cao hơn và việc phân bổ số lượng siêu cạnh cho các tầng nên tương ứng với kích thước đặc trưng của từng tầng.

e. **Chi phí của việc học cấu trúc siêu đồ thị:**

Tác giả so sánh thời gian chạy trung bình trên mỗi mẫu của ViT-Ti và Isotropic ViHGNN-Ti để đánh giá chi phí của việc học cấu trúc siêu đồ thị. Tác giả nhận định rằng việc gọi nhiều

lần hàm Fuzzy C-Means trong mô hình ViHGNN có thể dẫn đến chi phí tính toán lớn. Do đó, việc đo lường và so sánh thời gian chạy của ViHGNN với một mô hình không sử dụng Fuzzy C-Means như ViT là cần thiết. Kết quả sau khi thực nghiệm là như sau:

Table 10. Comparison of the mean running time (*ms*) per sample.

| Models | Forward | Backward | Total |
|---|---|---|---|
| ViT-Ti | 10.74 | 19.03 | 29.77 |
| Istropic ViHGNN-Ti | 11.25 | 18.77 | 30.02 |

Hình 11: Kết quả so sánh thời gian chạy trung bình trên mỗi mẫu của ViHGNN và ViT

Bảng trong hình 11 thể hiện thời gian chạy trung bình (tính bằng mili giây) cho mỗi mẫu của hai mô hình, được chia thành ba cột là:

- **Forward:** Thời gian lan truyền thuận, bao gồm cả chi phí của việc phân cụm bằng Fuzzy C-Means (đối với ViHGNN).
- **Backward:** Thời gian lan truyền ngược.
- **Total:** Tổng thời gian chạy.

Kết quả nhận được cho thấy:

- Thời gian lan truyền thuận của Isotropic ViHGNN-Ti chỉ tăng nhẹ so với ViT-Ti (11.25 ms so với 10.74 ms), cho thấy chi phí phân cụm bằng Fuzzy C-Means không đáng kể.
- Thời gian lan truyền ngược của hai mô hình gần như tương đương, chứng tỏ việc sử dụng Fuzzy C-Means không ảnh hưởng đến quá trình lan truyền ngược.
- Tóm lại, việc phân cụm bằng Fuzzy C-Means, mặc dù được gọi nhiều lần trong quá trình huấn luyện ViHGNN, không gây ra chi phí tính toán đáng kể.

**3. Áp dụng trên nhiệm vụ phát hiện đối tượng**

Ở phần này, tác giả áp dụng ViHGNN cho nhiệm vụ phát hiện đối tượng, trên tập dữ liệu COCO val2017, so sánh hiệu suất của mô hình Pyramid ViHGNN-S với các bộ xương (backbone) khác như ResNet50, ResNeXt-101-32x4d, PVT-Small, CycleMLP-B2, Swin-T và Pyramid ViG-S trên hai framework phát hiện đối tượng là RetinaNet và Mask R-CNN. Mục đích của là đánh giá khả năng tổng quát hóa của mô hình ViHGNN khi được áp dụng cho một nhiệm vụ thị giác khác ngoài phân loại ảnh. Kết quả sau thực nghiệm là:

Table 11. Object detection and instance segmentation results on COCO val2017.

| Backbone | RetinaNet 1× | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Params (M) | FLOPs (B) | mAP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
| ResNet50 [22] | 37.7 | 239.3 | 36.3 | 55.3 | 38.6 | 19.3 | 40.0 | 48.8 |
| ResNeXt-101-32x4d [68] | 56.4 | 319 | 39.9 | 59.6 | 42.7 | 22.3 | 44.2 | 52.5 |
| PVT-Small [62] | 34.2 | 226.5 | 40.4 | 61.3 | 44.2 | 25.0 | 42.9 | 55.7 |
| CycleMLP-B2 [5] | 36.5 | 230.9 | 40.6 | 61.4 | 43.2 | 22.9 | 44.4 | 54.5 |
| Swin-T [41] | 38.5 | 244.8 | 41.5 | 62.1 | 44.2 | 25.1 | 44.9 | 55.5 |
| Pyramid ViG-S [18] | 36.2 | 240.0 | 41.8 | 63.1 | 44.7 | 28.5 | 45.4 | 53.4 |
| Pyramid ViHGNN-S (ours) | 37.9 | 243.7 | **42.2** | **63.8** | **45.1** | **29.3** | **45.9** | **55.7** |
| Backbone | Mask R-CNN 1× | | | | | | | |
| | Param | FLOPs | $AP^b$ | $AP^b_{50}$ | $AP^b_{75}$ | $AP^m$ | $AP^m_{50}$ | $AP^m_{75}$ |
| ResNet50 [22] | 44.2 | 260.1 | 38.0 | 58.6 | 41.4 | 34.4 | 55.1 | 36.7 |
| PVT-Small [62] | 44.1 | 245.1 | 40.4 | 62.9 | 43.8 | 37.8 | 60.1 | 40.3 |
| CycleMLP-B2 [5] | 46.5 | 249.5 | 42.1 | 64.0 | 45.7 | 38.9 | 61.2 | 41.8 |
| PoolFormer-S24 [74] | 41.0 | - | 40.1 | 62.2 | 43.4 | 37.0 | 59.1 | 39.6 |
| Swin-T [41] | 47.8 | 264.0 | 42.2 | 64.6 | 46.2 | 39.1 | 61.6 | 42.0 |
| Pyramid ViG-S [18] | 45.8 | 258.8 | 42.6 | 65.2 | 46.0 | 39.4 | 62.4 | 41.6 |
| Pyramid ViHGNN-S (ours) | 47.3 | 261.4 | **43.1** | **66.0** | **46.5** | **39.6** | **63.0** | **42.3** |

Hình 12: Kết quả trên nhiệm vụ phát hiện đối tượng

Kết quả nhận được từ hình 12 cho thấy:

- **Đối với RetinaNet:** Pyramid ViHGNN-S đạt được 42.2% mAP, cao hơn so với các bộ xương sống đại diện khác, bao gồm cả Pyramid ViG-S (41.8% mAP). Điều này cho thấy lợi thế của việc sử dụng siêu đồ thị để biểu diễn hình ảnh trong nhiệm vụ phát hiện đối tượng.

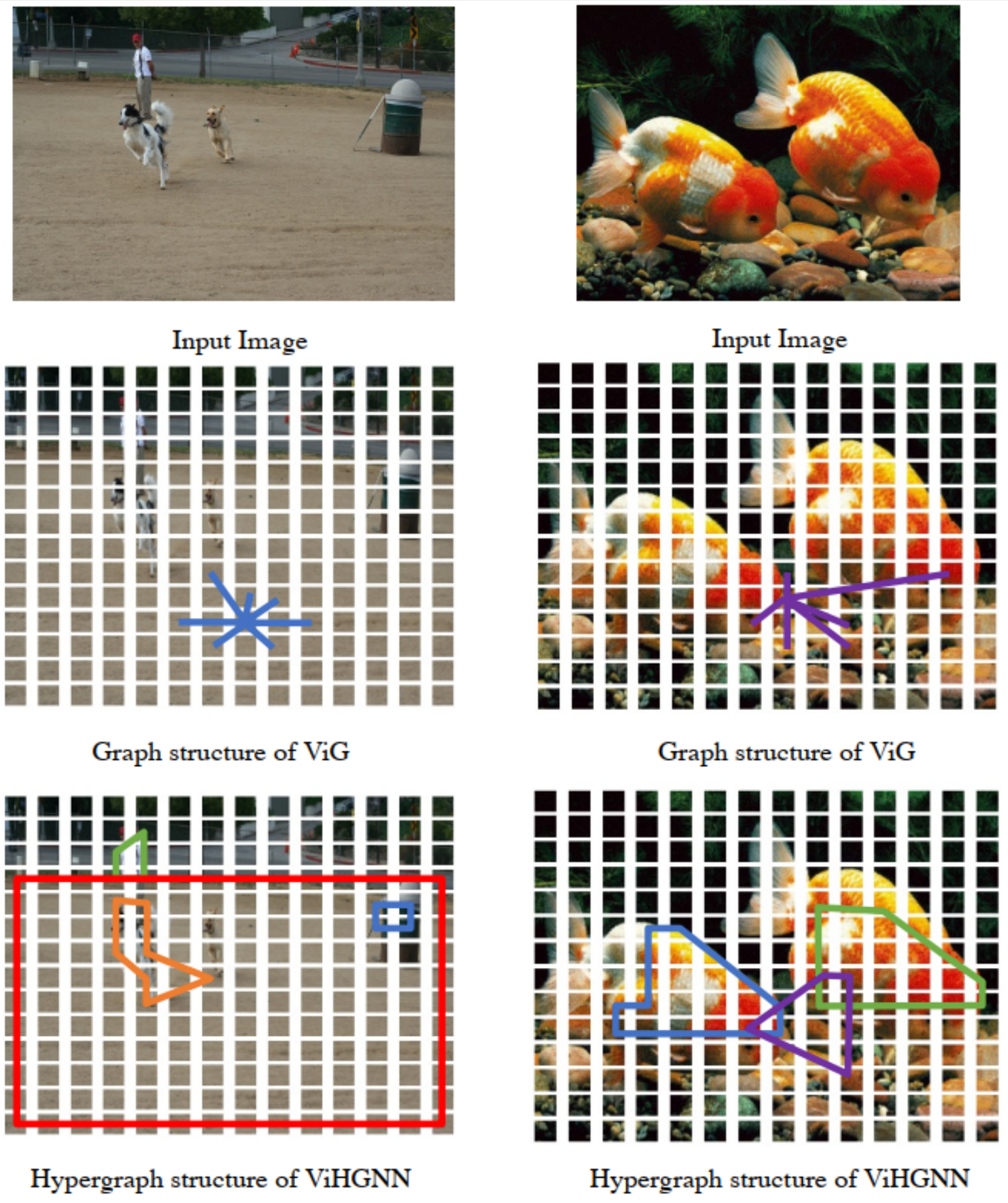
- **Đối với Mask R-CNN:** Tương tự như RetinaNet, Pyramid ViHGNN-S cũng vượt trội hơn so với các bộ xương khác trên Mask R-CNN với 43.1% $AP^b$, cao hơn Pyramid ViG-S (42.6% $AP^b$).
- Nhìn chung, Pyramid ViHGNN-S đạt được hiệu suất tốt hơn so với các bộ xương khác trên cả hai framework RetinaNet và Mask R-CNN, chứng minh khả năng khái quát hóa và hiệu quả của kiến trúc ViHGNN trong nhiệm vụ phát hiện đối tượng.

# 8 Kết Luận

**So sánh Vision GNN với Vision HGNN**

Ở phần này tác giả cung cấp một minh họa trực quan về cách mô hình ViHGNN hiểu và xử lý thông tin hình ảnh so với mô hình ViG.

Hình 13 bên dưới minh họa trực quan cho sự khác biệt giữa cấu trúc đồ thị của ViG và cấu trúc hypergraph của ViHGNN.




Hình 13: So sánh ViG với ViHGNN

Từ hình ảnh trực quan, có thể thấy rằng:

- **ViG có xu hướng tạo ra các cạnh dư thừa** giữa các mảng có đặc trưng cục bộ giống nhau như màu sắc và kết cấu (ví dụ: các vùng có các mảng giống nhau, chẳng hạn như cát trong hình ảnh bên trái). Sự dư thừa này thể hiện sự lãng phí tài nguyên mà ViHGNN giải quyết bằng cách sử dụng ít siêu cạnh hơn để mô hình hóa các mối quan hệ như vậy. Ngược lại, **ViHGNN sử dụng các siêu cạnh để nắm bắt các mối quan hệ bậc cao hơn giữa các mảng**, cho phép biểu diễn hiệu quả hơn các vùng có kết cấu tương tự. Thay vì tạo ra nhiều cạnh dư thừa, ViHGNN sẽ nhóm các mảng cát tương tự thành một siêu cạnh, giảm thiểu lãng phí tài nguyên và nắm bắt hiệu quả hơn các mối quan hệ phức tạp.
- **ViG cũng có thể tạo ra các cạnh nhiễu**, kết nối các mảng đối tượng với các mảng khác biệt về mặt ngữ nghĩa nhưng có các thuộc tính cục bộ tương tự (như màu sắc). Ví dụ, trong hình 13 con cá, ViG có thể tạo ra nhiều cạnh dư thừa giữa liên kết các mảng màu của 2 con cá khác nhau, trong khi ViHGNN sẽ sử dụng ít siêu cạnh hơn để biểu diễn mối quan hệ giữa các con cá này.
- ViHGNN sử dụng các siêu cạnh để nắm bắt các mối quan hệ bậc cao hơn giữa các mảng

Tóm lại, phần trực quan hóa cho thấy ViHGNN học được cấu trúc hypergraph hiệu quả và ý nghĩa hơn so với cấu trúc đồ thị của ViG, cho thấy khả năng học biểu diễn hình ảnh nâng cao của ViHGNN.

#### Thực nghiệm và đánh giá

- **ViHGNN trên nhiệm vụ phân loại ảnh:**

  ViHGNN đã chứng minh hiệu suất vượt trội trong phân loại ảnh trên ImageNet, đặc biệt so với các mô hình isotropic và pyramid khác, nhờ sử dụng siêu đồ thị. Mô hình này cải thiện độ chính xác và duy trì chi phí tính toán tương đương, khẳng định tiềm năng trong các tác vụ thị giác máy tính bằng cách nắm bắt mối quan hệ phức tạp giữa các phần tử hình ảnh.

- **ViHGNN trên nhiệm vụ phát hiện đối tượng:**

  ViHGNN cũng đã được áp dụng thành công cho nhiệm vụ phát hiện đối tượng, chứng tỏ khả

năng tổng quát hóa của mô hình. Khi được sử dụng làm xương sống cho các khung phát hiện RetinaNet và Mask R-CNN, ViHGNN đã đạt được hiệu suất vượt trội so với các xương sống khác trên bộ dữ liệu COCO.

- **Ảnh hưởng của từng thành phần trong mô hình ViHGNN:**

  Các nghiên cứu, thử nghiệm trên ViHGNN cho thấy Fuzzy C-Means là phương pháp xây dựng siêu đồ thị vượt trội, nhờ khả năng nắm bắt các mối quan hệ phức tạp trong nhúng bản vá với chi phí thấp hơn so với K-Means và các phương pháp phân cụm khác. Một vòng lặp cập nhật là đủ để duy trì hiệu suất tối ưu cho ViHGNN-Ti, vì thêm vòng lặp không mang lại cải thiện đáng kể. Số lượng siêu cạnh lý tưởng cần được điều chỉnh sao cho phù hợp với số lượng nút bản vá, nhằm cân bằng giữa hiệu suất và chi phí tính toán. Cuối cùng, module HSL cải thiện hiệu suất của mô hình với chi phí rất nhỏ về kích thước và FLOPs, không gây ra chi phí tính toán đáng kể. Tóm lại, Fuzzy C-Means, số lượng siêu cạnh tối ưu, và module HSL là các yếu tố quan trọng trong việc tối ưu hóa ViHGNN.

# Tài liệu tham khảo

[1] Hypergraph neural networks for hypergraph matching. (2021, October 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9710894

[2] Emami. (2023, March 12). History of Graph Neural Networks (GNN) - Emami - Medium. https://medium.com/@saluem/graph-neural-networks-gnn-93b32567a6d9