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

BÁO CÁO
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 Group Information 3

2 Thông tin nhóm 3

3 Current Status 4

3.1 Tiến độ công việc 4

3.2 Khó khăn gặp phải và Cách giải quyết 4

4 Introduction 6

4.1 Bối cảnh và vấn đề 6

4.2 Động lực 6

4.3 Ý nghĩa Khoa học và Ứng dụng Thực tế của ViG 8

5 Preliminaries and Backgrounds 10

5.1 Notation and Definitions 10

5.2 Problem Statement 11

5.3 General Frameworks 12

5.4 Thách thức và Hạn chế 13

6 Related Works 16

7 Methodology 19

7.1 Biểu diễn ảnh dưới dạng đồ thị 19

7.2 Cấu trúc ViG Block 20

7.3 Kiến trúc mạng ViG 22

7.4 Mã hóa vị trí 24

7.5 Ưu điểm 24
7.6 Nhược điểm 25
8 Experimental & Evaluations 26
8.1 Bộ dữ liệu và Thiết lập Thực nghiệm 26
8.2 Chỉ số Đánh giá 27
8.3 Kết quả Chính trên ImageNet 28
8.4 Phát hiện Đối tượng trên COCO 32
9 Ablation Study 33
9.1 So sánh các loại Graph Convolution 33
9.2 Đánh giá tác động của các Module 34
9.3 Ảnh hưởng của số lượng Node Láng Giềng (K) 35
9.4 Đánh giá số lượng Heads trong Multi-Head Attention 35
9.5 Kết luận từ Nghiên cứu Ablation 36
10 Conclusion 37
10.1 Tổng kết các phát hiện chính 37
10.2 Đóng góp và ý nghĩa 37
10.3 Hạn chế và hướng phát triển tương lai 38
Tài liệu tham khảo 39

1 Group Information

2 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. • Chuẩn bị slide thuyết trình về các nội dung đã tìm hiểu. • Viết báo cáo. • 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. • Chuẩn bị slide thuyết trình về nội dung liên quan. • Viết báo cáo. • Thuyết trình buổi Seminar.	100%
21127229	Dương Trường Bình	• Đọc paper Vision GNN và tìm hiểu về phần Related Works, Experimental & Evaluations và Conclusion. • Kiểm thử lại code về GNN và xem xét kết quả kiểm thử. • Viết báo cáo. • Chuẩn bị slide thuyết trình về các nội dung liên quan.. • 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. • Viết báo cáo. • Chuẩn bị slide thuyết trình về các nội dung đã tìm hiểu. • Thuyết trình buổi Seminar.	100%

3 Current Status

3.1 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

3.2 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.

4 Introduction

4.1 Bối cảnh và vấn đề

Trong lĩnh vực thị giác máy tính, các kiến trúc mạng nơ-ron sâu chủ yếu dựa trên ba loại chính: mạng nơ-ron tích chập (CNN), Transformer, và mạng MLP. Mỗi loại kiến trúc này đã đạt được những thành công đáng kể trong các tác vụ thị giác như phân loại hình ảnh (image classification), phát hiện đối tượng (objec detection) và phân đoạn ngữ nghĩa (sematic segmentation). Tuy nhiên, các kiến trúc này có những hạn chế riêng và thường không thể khai thác hết các mối quan hệ phức tạp trong dữ liệu hình ảnh: CNN xử lý hình ảnh dưới dạng lưới, Transformer xử lý dưới dạng chuỗi, điều này không linh hoạt để nắm bắt các đối tượng phức tạp và không đều trong các tác vụ thị giác. Mô hình Vision GNN (ViG) chuyển đổi hình ảnh thành đồ thị, với mỗi patch của hình ảnh được coi là một nút đồ thị, từ đó sử dụng các module Grapher và FFN để xử lý thông tin. Chủ đề này không chỉ mang tính lý thuyết cao mà còn mở ra những hướng nghiên cứu và ứng dụng mới trong thị giác máy tính.

4.2 Động lực

Động lực nghiên cứu chủ đề này xuất phát từ nhu cầu khắc phục những hạn chế của các kiến trúc mạng nơ-ron hiện tại và khai thác tối đa tiềm năng của mạng nơ-ron đồ thị trong các tác vụ thị giác. Cụ thể:

- **Hạn chế của CNN và Transformer trong việc xử lý các đối tượng phức tạp**: Các mạng nơ-ron tích chập (CNN) và Transformer, những kiến trúc thống trị trong thị giác máy tính, coi hình ảnh là cấu trúc lưới hoặc chuỗi. Cách tiếp cận này tỏ ra kém hiệu quả khi xử lý các đối tượng có hình dạng bất thường và phức tạp vì nó không linh hoạt trong việc nắm bắt các mối quan hệ không gian phức tạp giữa các phần của đối tượng
- **Tính linh hoạt của cấu trúc đồ thị**: Biểu diễn hình ảnh dưới dạng đồ thị, với các nút đại diện cho các phần của hình ảnh và các cạnh thể hiện mối quan hệ giữa chúng, mang lại sự linh hoạt

cao hơn trong việc mô hình hóa các đối tượng phức tạp. Cấu trúc đồ thị cho phép nắm bắt các mối quan hệ không gian dài hạn giữa các phần của đối tượng một cách hiệu quả hơn. Điều này đặc biệt hữu ích khi xử lý các đối tượng có hình dạng bất thường hoặc phức tạp, nơi mà các phương pháp truyền thống như CNN và Transformer thường không đạt được hiệu quả cao.

- **Tiềm năng của GNN trong thị giác máy tính**: Các nghiên cứu về GNN đã cho thấy hiệu quả trong việc xử lý dữ liệu đồ thị trong các lĩnh vực khác nhau như mạng xã hội, mạng trích dẫn và hóa sinh. Tuy nhiên, việc áp dụng GNN trong thị giác máy tính còn hạn chế, chủ yếu tập trung vào các tác vụ cụ thể với cấu trúc đồ thị tự nhiên như phân loại đám mây điểm (point clouds classification), tạo đồ thị cảnh (scene graph generation) và nhận dạng hành động (action recognition). Nghiên cứu này nhằm mục đích khai thác tiềm năng của GNN bằng cách đề xuất một kiến trúc dựa trên GNN có thể xử lý trực tiếp dữ liệu hình ảnh cho các tác vụ thị giác máy tính nói chung. Việc này không chỉ mở ra hướng nghiên cứu mới mà còn có thể mang lại những tiến bộ đáng kể trong việc giải quyết các vấn đề thị giác máy tính phức tạp.

Tác động và Ý nghĩa Tiềm năng

Nghiên cứu này có tiềm năng mang lại những tác động và ý nghĩa quan trọng đối với lĩnh vực thị giác máy tính:

- **Mở ra hướng đi mới cho kiến trúc mạng nơ-ron**: Việc sử dụng GNN để biểu diễn và xử lý hình ảnh có thể dẫn đến sự phát triển của các kiến trúc mạng nơ-ron mới hiệu quả và mạnh mẽ hơn cho thị giác máy tính.
- **Nâng cao hiệu suất cho các tác vụ thị giác máy tính**: Cấu trúc đồ thị linh hoạt của Vision GNN có tiềm năng cải thiện hiệu suất cho các tác vụ thị giác máy tính khác nhau, đặc biệt là những tác vụ liên quan đến các đối tượng phức tạp và bất thường.
- **Mở rộng ứng dụng của GNN**: Nghiên cứu này có thể thúc đẩy việc áp dụng GNN trong một loạt các ứng dụng thị giác máy tính, vượt ra ngoài các tác vụ cụ thể đã được khám phá trước đây.

Tóm lại, nghiên cứu về Vision GNN được thúc đẩy bởi những hạn chế của các kiến trúc mạng nơ-ron hiện có và tiềm năng của GNN trong việc xử lý dữ liệu hình ảnh phức tạp. Nghiên cứu này có tiềm năng mang lại những tác động và ý nghĩa quan trọng đối với lĩnh vực thị giác máy tính.

4.3 Ý nghĩa Khoa học và Ứng dụng Thực tế của ViG

Ý nghĩa Khoa học

- ViG là một kiến trúc mạng mới trong lĩnh vực thị giác máy tính, sử dụng đồ thị để biểu diễn hình ảnh thay vì cấu trúc lưới hoặc chuỗi truyền thống. Cách tiếp cận này mang tính sáng tạo và mở ra hướng nghiên cứu mới cho việc xử lý hình ảnh dựa trên đồ thị.
- Việc biểu diễn hình ảnh dưới dạng đồ thị mang lại nhiều lợi thế, bao gồm tính linh hoạt trong việc mô hình hóa các đối tượng phức tạp, khả năng bắt giữ mối quan hệ giữa các phần của đối tượng và tận dụng các nghiên cứu tiên tiến về GNN.
- ViG giải quyết vấn đề over-smoothing trong các GNN sâu bằng cách giới thiệu các phép biến đổi đặc trưng và các hàm kích hoạt phi tuyến tính. Điều này cho phép ViG duy trì tính đa dạng của đặc trưng và học được các biểu diễn phân biệt
- **Khắc phục Hạn chế của CNN và Transformer**: Các mô hình CNN và Transformer, mặc dù thành công, nhưng có những hạn chế trong việc xử lý các đối tượng có hình dạng bất thường. ViG, bằng cách sử dụng biểu diễn đồ thị, cung cấp một giải pháp thay thế linh hoạt và hiệu quả hơn
- **Mở ra Hướng Nghiên cứu Mới**: Là một trong những nghiên cứu tiên phong áp dụng GNN cho các tác vụ thị giác quy mô lớn, ViG mở ra hướng nghiên cứu mới đầy hứa hẹn trong lĩnh vực thị giác máy tính.

Ứng dụng Thực tế:

- ViG có thể được ứng dụng trong nhiều tác vụ thị giác máy tính khác nhau, bao gồm:

  - Nhận dạng hình ảnh: Các thí nghiệm trên tập dữ liệu ImageNet cho thấy ViG đạt được độ chính xác vượt trội so với các kiến trúc mạng khác như CNN, MLP và Transformer. Qua đó chứng tỏ khả năng của nó trong việc học các biểu diễn phân biệt cho nhận dạng hình ảnh.
  - Phát hiện đối tượng: ViG cũng cho thấy khả năng khái quát hóa tốt khi được sử dụng làm xương sống (backbone) cho các khung phát hiện đối tượng như RetinaNet và Mask R-CNN.

- Các ứng dụng tiềm năng khác của ViG bao gồm:
  - Phân đoạn hình ảnh: Khả năng biểu diễn đồ thị của ViG có thể hữu ích trong việc phân đoạn các đối tượng có hình dạng phức tạp.
  - Tạo chú thích hình ảnh: Việc nắm bắt mối quan hệ giữa các đối tượng trong hình ảnh có thể giúp tạo ra các chú thích chính xác hơn.

Tóm lại, việc biểu diễn hình ảnh dưới dạng đồ thị và sử dụng GNN như ViG mang đến một hướng tiếp cận mới đầy hứa hẹn cho thị giác máy tính. ViG không chỉ khắc phục những hạn chế của các kiến trúc mạng truyền thống mà còn mở ra những khả năng mới cho các ứng dụng thực tế trong tương lai.

5 Preliminaries and Backgrounds

5.1 Notation and Definitions

Để hiểu rõ hơn về cách thức hoạt động của mô hình ViG, trước hết chúng ta cần làm rõ một số khái niệm cơ bản về đồ thị và GNN. Trong phần này giới thiệu các ký hiệu và định nghĩa quan trọng:

- $G = (V, E)$: Đồ thị biểu diễn ảnh, trong đó V là tập các đỉnh và E là tập các cạnh.
- $v_i \in V$: Đỉnh thứ i trong đồ thị, tương ứng với một patch ảnh.
- $x_i \in \mathbb{R}^D$: Vector đặc trưng của đỉnh v_i, với D là số chiều của vector đặc trưng.
- $X = [x_1, x_2, ..., x_N] \in \mathbb{R}^{N \times D}$: Ma trận đặc trưng của toàn bộ đồ thị, với N là số lượng đỉnh.
- $e_{ji} \in E$: Cạnh nối từ đỉnh v_j đến đỉnh v_i.
- $\mathcal{N}(v_i)$: Tập hợp các đỉnh láng giềng của đỉnh v_i.
- K: Số lượng láng giềng gần nhất được xét cho mỗi đỉnh.
- $H \times W \times 3$: Kích thước của ảnh đầu vào, với H là chiều cao, W là chiều rộng.
- **Graph Neural Network (GNN)**: Một lớp các mô hình học sâu được thiết kế để xử lý dữ liệu có cấu trúc đồ thị. GNN có khả năng học và trích xuất đặc trưng từ các mối quan hệ giữa các đỉnh trong đồ thị.
- **Graph Convolution**: Phép toán tổng hợp thông tin từ các đỉnh láng giềng trong đồ thị, tương tự như phép tích chập trong CNN nhưng áp dụng trên cấu trúc đồ thị.
- **Image Patch**: Một phần nhỏ của ảnh, thường là một ô vuông, được sử dụng để chia nhỏ ảnh thành các đơn vị xử lý.
- **Feature Transformation**: Quá trình biến đổi vector đặc trưng của các đỉnh trong đồ thị nhằm tăng cường khả năng biểu diễn của mô hình.
- **Multi-head Update**: Kỹ thuật cập nhật đặc trưng của đỉnh bằng cách sử dụng nhiều bộ tham số

độc lập, giúp mô hình học được nhiều biểu diễn khác nhau.

- **Oversmoothing**: Hiện tượng các vector đặc trưng của các đỉnh trong đồ thị trở nên quá giống nhau sau nhiều lớp GNN, làm giảm khả năng phân biệt của mô hình.
- **Isotropic Architecture**: Kiến trúc mạng trong đó kích thước đặc trưng được giữ nguyên qua các lớp.
- **Pyramid Architecture**: Kiến trúc mạng trong đó kích thước đặc trưng giảm dần qua các lớp, thường đi kèm với việc tăng số kênh đặc trưng.
- **Positional Encoding**: Kỹ thuật mã hóa thông tin vị trí của các đỉnh trong đồ thị, giúp mô hình nhận biết được cấu trúc không gian của ảnh.

5.2 Problem Statement

Nghiên cứu này nhắm đến giải quyết ba vấn đề chính trong lĩnh vực thị giác máy tính:

Hạn chế của CNN và Transformer trong xử lý đối tượng phức tạp

Các kiến trúc mạng hiện tại như CNN và Transformer thường xử lý ảnh dưới dạng lưới hoặc chuỗi, điều này gây ra những hạn chế nhất định:

> "Since the objects are usually not quadrate whose shape is irregular, the commonly-used grid or sequence structures in previous networks like ResNet and ViT are redundant and inflexible to process them."

Cấu trúc lưới và chuỗi không phù hợp để biểu diễn các đối tượng có hình dạng bất thường, dẫn đến việc xử lý dư thừa và thiếu linh hoạt.

Nhu cầu về một cấu trúc linh hoạt hơn

Đồ thị được đề xuất như một giải pháp để biểu diễn dữ liệu hình ảnh một cách linh hoạt và hiệu quả hơn:

> "An object can be viewed as a composition of parts, e.g., a human can be roughly divided into head, upper body, arms and legs. These parts linked by joints naturally form a graph structure. By analyzing the graph, we are able to recognize the human. Moreover, **graph is a generalized data structure that grid and sequence can be viewed as a special case of graph**. Viewing an image as a graph is more flexible and effective for visual perception."

Cấu trúc đồ thị có khả năng nắm bắt các mối quan hệ phức tạp giữa các thành phần của đối tượng, đồng thời cũng là một cấu trúc dữ liệu tổng quát hơn so với lưới và chuỗi.

Khai thác tiềm năng của GNN trong thị giác máy tính

Mặc dù GNN đã được áp dụng trong một số nhiệm vụ cụ thể, nhưng tiềm năng của nó như một kiến trúc chính cho xử lý ảnh vẫn chưa được khai thác đầy đủ:

> "GCN can only tackle specific visual tasks with naturally constructed graph. For general applications in computer vision, we need a GCN-based backbone network that directly processes the image data."

Để giải quyết các vấn đề trên, các tác giả đề xuất một kiến trúc mạng mới - Vision GNN (ViG), nhằm tận dụng sức mạnh của biểu diễn đồ thị để xử lý hiệu quả và linh hoạt hơn các dữ liệu hình ảnh phức tạp

5.3 General Frameworks

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.
  - **CNN:** Sử dụng ma trận pixel trực tiếp.
  - **Vision Transformer:** Chia ảnh thành các patch và biểu diễn dưới dạng chuỗi.

  - **ViG:** Biểu diễn ảnh dưới dạng đồ thị với các patch là các đỉnh.

- **Trích xuất đặc trưng:** Sử dụng các kiến trúc mạng để học và trích xuất các đặc trưng từ ảnh.
  - **CNN:** Sử dụng các lớp tích chập (convolution layers).
  - **Vision Transformer:** Sử dụng cơ chế self-attention.
  - **ViG:** Sử dụng graph convolution và FFN modules.
- **Tổng hợp thông tin:** Kết hợp thông tin từ các đặc trưng đã trích xuất.
  - **CNN:** Sử dụng pooling layers.
  - **Vision Transformer:** Sử dụng token đặc biệt (CLS token).
  - **ViG:** Sử dụng graph pooling hoặc adaptive average pooling.
- **Phân loại hoặc dự đoán:** Sử dụng các lớp fully connected hoặc MLP để đưa ra kết quả cuối cùng.

Mỗi phương pháp có cách tiếp cận riêng trong việc thực hiện các bước này, nhưng đều nhằm mục đích tối ưu hóa khả năng học và trích xuất đặc trưng từ dữ liệu ảnh. ViG đề xuất một cách tiếp cận mới bằng cách kết hợp ưu điểm của GNN với các kỹ thuật xử lý ảnh, nhằm cải thiện hiệu suất và tính linh hoạt của mô hình trong các tác vụ thị giác máy tính.

5.4 Thách thức và Hạn chế

Mô hình Vision GNN (ViG) đề xuất một cách tiếp cận mới trong việc xử lý ảnh bằng cách sử dụng Graph Neural Networks. Tuy nhiên, phương pháp này cũng đối mặt với một số thách thức và hạn chế:

a. **Vấn đề Over-smoothing:**

- Hiện tượng over-smoothing là một thách thức lớn trong các mô hình GNN sâu.

- Khi số lượng lớp GNN tăng lên, các vector đặc trưng của các đỉnh có xu hướng trở nên quá giống nhau.
- Điều này làm giảm khả năng phân biệt của mô hình, đặc biệt là trong các tác vụ nhận dạng hình ảnh phức tạp.

b. **Tính toán phức tạp:**

- Việc xây dựng và cập nhật đồ thị cho mỗi ảnh có thể tốn nhiều tài nguyên tính toán.
- Đặc biệt với ảnh có độ phân giải cao, số lượng đỉnh và cạnh trong đồ thị có thể rất lớn.
- Cần có các phương pháp tối ưu hóa để giảm độ phức tạp tính toán mà vẫn duy trì hiệu suất của mô hình.

c. **Biểu diễn thông tin không gian:**

- Mặc dù ViG sử dụng positional encoding, việc duy trì và khai thác hiệu quả thông tin không gian của ảnh trong cấu trúc đồ thị vẫn là một thách thức.
- Cần có các phương pháp tiên tiến hơn để tích hợp thông tin vị trí vào quá trình học của mô hình.

d. **Khả năng mở rộng:**

- Hiệu suất của ViG trên các tập dữ liệu lớn hơn và đa dạng hơn cần được nghiên cứu thêm.
- Việc áp dụng mô hình cho các tác vụ thị giác máy tính phức tạp hơn như phân đoạn ảnh hay phát hiện đối tượng vẫn cần được khám phá sâu hơn.

e. **Thiếu các kiến trúc chuẩn:**

- Không giống như CNN hay Transformer, GNN cho xử lý ảnh chưa có các kiến trúc chuẩn được công nhận rộng rãi.
- Cần có thêm nghiên cứu để xây dựng các kiến trúc GNN hiệu quả và có thể tái sử dụng cho nhiều tác vụ thị giác máy tính khác nhau.

Những thách thức và hạn chế này mở ra nhiều hướng nghiên cứu tiềm năng trong tương lai. Việc giải quyết các vấn đề này có thể đưa đến sự phát triển của các mô hình GNN mạnh mẽ hơn cho xử lý ảnh, mang lại những tiến bộ đáng kể trong lĩnh vực thị giác máy tính.

6 Related Works

2012: Sự Ra Đời của CNN (Convolutional Neural Networks)

- **Khởi đầu với LeNet:** Đánh dấu bước ngoặt quan trọng khi CNN được sử dụng thành công trong nhiều bài toán thị giác máy tính.
- **Đặc điểm chính:** Sử dụng các lớp tích chập áp dụng sliding window để trích xuất đặc trưng không gian từ ảnh, mang lại hiệu suất vượt trội so với các phương pháp truyền thống.
- **Phát triển tiếp theo:** Từ đó nhiều kiến trúc mạng CNN phức tạp hơn đã ra đời, như VGGNet, ResNet, và Inception, tiếp tục cải thiện hiệu suất và khả năng ứng dụng của CNN.

2017: Sự Chú Ý Đến GNN (Graph Neural Networks)

- GNN bắt đầu thu hút sự chú ý trong cộng đồng nghiên cứu với mục tiêu xử lý dữ liệu có cấu trúc đồ thị.
- Graph Convolutional Networks (GCN) mở ra tiềm năng ứng dụng của GNN trong nhiều lĩnh vực:
  - **Graph Convolutional Networks (GCN):** GCN được thiết kế để áp dụng phép biến đổi tích chập trên các đồ thị, thay vì trên pixel như trong CNN. Phép tích chập đồ thị giúp tổng hợp thông tin từ các nút láng giềng và cập nhật đặc trưng của nút hiện tại. GCN thường áp dụng trên các loại dữ liệu có cấu trúc đồ thị như mạng xã hội, đồ thị phân tử, và dữ liệu điểm 3D.
  - **Nhược điểm:** Khi số lượng lớp tích chập trong GCN tăng lên, các đặc trưng của các nút khác nhau trong đồ thị có xu hướng trở nên giống nhau, dẫn đến hiện tượng over-smoothing, làm cho mô hình mất đi khả năng phân biệt chi tiết giữa các nút khác nhau.

2017: Sự Xuất Hiện của Transformer

- Transformer được giới thiệu trong bài báo "Attention is All You Need", và nhanh chóng thành công trong xử lý ngôn ngữ tự nhiên (NLP).

- **Đặc điểm chính:** Transformer không dựa trên cấu trúc tuần tự truyền thống mà sử dụng cơ chế self-attention, cho phép mô hình tập trung vào các phần quan trọng của dữ liệu đầu vào và học được mối quan hệ dài hạn giữa các thành phần.

2020: Vision Transformer (ViT)

- Vision Transformer (ViT) đánh dấu bước tiến quan trọng khi áp dụng Transformer vào thị giác máy tính.
- **Đặc điểm chính:**
  - ViT chia hình ảnh đầu vào thành các patch kích thước cố định, sau đó ánh xạ mỗi patch thành vector đặc trưng thông qua một lớp embedding. Các vector đặc trưng từ các patch được đưa vào lớp self-attention, cho phép mô hình nắm bắt mối quan hệ giữa các patch khác nhau trên toàn bộ ảnh.
  - **Ưu điểm:** ViT có khả năng nắm bắt mối quan hệ dài hạn giữa các vùng của ảnh và đạt hiệu suất cao trên nhiều bộ dữ liệu lớn như ImageNet.
  - **Nhược điểm:** ViT yêu cầu lượng dữ liệu lớn và tài nguyên tính toán mạnh cũng như cần sự tối ưu hóa kỹ thuật cao để đạt được hiệu suất tốt nhất.

2020-2022: Phát Triển Các Kiến Trúc Dựa Trên MLP

- Khoảng thời gian này chứng kiến sự phát triển của các kiến trúc dựa trên MLP, không sử dụng lớp tích chập hay cơ chế self-attention của Transformer.
- **Đại diện:** Các mô hình như CycleMLP, ResMLP đã chứng minh rằng MLP cũng có thể đạt được hiệu suất cạnh tranh trong các nhiệm vụ thị giác máy tính, nhờ vào các thiết kế module đặc biệt.
  - **Ưu điểm:** Đơn giản hơn trong thiết kế và có thể được tối ưu hóa tốt trên nhiều nền tảng phần cứng khác nhau.
  - **Nhược điểm:** MLP có thể gặp khó khăn khi xử lý các thông tin phức tạp và không gian cao

hơn do thiếu khả năng trích xuất đặc trưng không gian sâu sắc như CNN hay Transformer.

2022: Phát Triển Các Mô Hình ViG (Vision GNN)

- **Giới thiệu:** Vision GNN (ViG) là mô hình đầu tiên áp dụng GNN vào thị giác máy tính, sử dụng các nút và cạnh để mô hình hóa mối quan hệ không gian trong ảnh.
- **Đặc điểm chính:** Chia hình ảnh thành các patch, mỗi patch tương ứng với một node và sử dụng graph convolution để tổng hợp thông tin và cập nhật đặc trưng cho các node.

ViG được phát triển dựa trên GCN (Graph Convolutional Networks) kết hợp với các cải tiến trong lĩnh vực thị giác máy tính để áp dụng vào hình ảnh. ViG không phải là một hướng hoàn toàn mới, nhưng là sự kết hợp và phát triển từ các ý tưởng trước đó trong GNN và thị giác máy tính.

7 Methodology

7.1 Biểu diễn ảnh dưới dạng đồ thị

Đầu vào của mô hình Vision GNN (ViG) là một hình ảnh có kích thước $H \times W \times 3$. Quá trình biến đổi hình ảnh thành đồ thị gồm các bước sau:

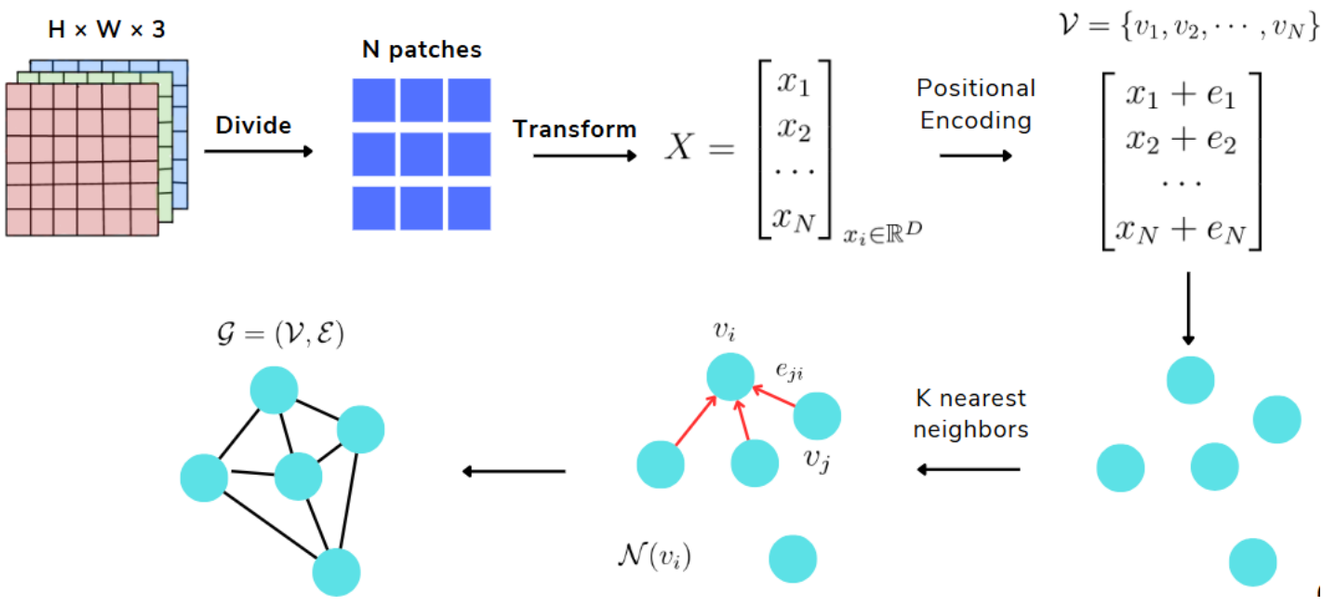
a. Chia hình ảnh thành N patch nhỏ hơn.

b. Mỗi patch được chuyển đổi thành một vector đặc trưng $\mathbf{x}_i \in \mathbb{R}^D$, với $i = 1, 2, ..., N$.

c. Bổ sung thông tin vị trí cho mỗi vector đặc trưng: $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{e}_i$, với $\mathbf{e}_i \in \mathbb{R}^D$ là vector mã hóa vị trí.

d. Tạo tập hợp các node $V = \{v_1, v_2, ..., v_N\}$, với v_i tương ứng với x_i.

e. Xây dựng đồ thị $G = (V, E)$ bằng cách kết nối mỗi node với K node lân cận gần nhất và tạo cạnh có hướng e_{ji} từ $\mathbf{v}_j$ đến $\mathbf{v}_i$ với mọi $\mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i)$.



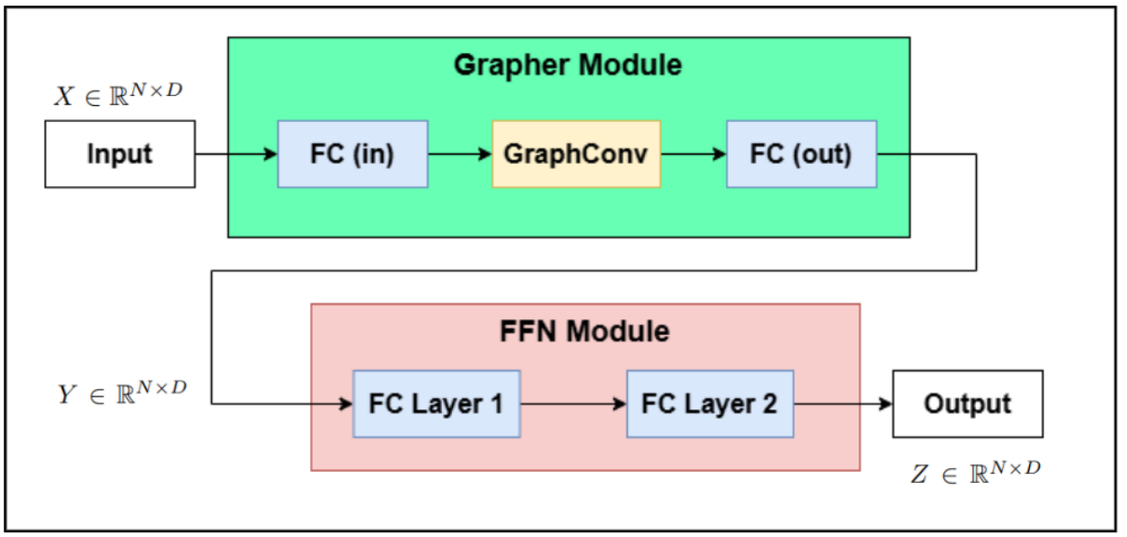
Công thức biểu diễn đồ thị:

$$G = (V, E), \quad V = \{v_1, v_2, ..., v_N\}, \quad E = \{e_{ji} | v_j \in \mathcal{N}(v_i)\} \tag{1}$$

trong đó $\mathcal{N}(v_i)$ là tập hợp K node lân cận gần nhất của v_i.

7.2 Cấu trúc ViG Block

ViG Block là đơn vị cơ bản của mạng Vision GNN, bao gồm hai module chính: Grapher Module và FFN Module.



Grapher Module

Grapher Module thực hiện phép tích chập trên đồ thị để học mối quan hệ giữa các node trong đồ thị ảnh, bao gồm các thành phần sau:

- **Lớp FC đầu vào:** Chiếu đặc trưng node vào không gian mới.
- **GraphConv:** Thực hiện phép tích chập trên đồ thị.
- **Lớp FC đầu ra:** Biến đổi đặc trưng sau tích chập.

Công thức của Grapher Module:

$$Y = \sigma(\text{GraphConv}(XW_{in}))W_{out} + X \tag{2}$$

trong đó:

- $\mathbf{X} \in \mathbb{R}^{N \times D}$ là ma trận đầu vào
- $\mathbf{W}_{\text{in}}$ và $\mathbf{W}_{\text{out}}$ là các ma trận trọng số học được
- σ là hàm kích hoạt phi tuyến (ví dụ: ReLU, GeLU)
- GraphConv là phép tích chập trên đồ thị sử dụng phương pháp Max-Relative Graph Convolution, được định nghĩa như sau:

$$x_i' = h(x_i, g(x_i, \mathcal{N}(x_i), W_{agg}), W_{update}) \tag{3}$$

với với $g(\cdot)$ là hàm tổng hợp thông tin từ các node lân cận và $h(\cdot)$ là hàm cập nhật node:

$$g(\cdot) = x_i'' = [x_i, \max(\{x_j - x_i | j \in \mathcal{N}(x_i)\})] \tag{4}$$

$$h(\cdot) = x_i' = x_i'' W_{update} \tag{5}$$

FFN (Feed-Forward Network) Module

FFN Module tăng cường khả năng biểu diễn phi tuyến của mỗi node và giúp giảm hiện tượng over-smoothing. Cấu trúc gồm hai lớp Fully Connected với hàm kích hoạt phi tuyến ở giữa:

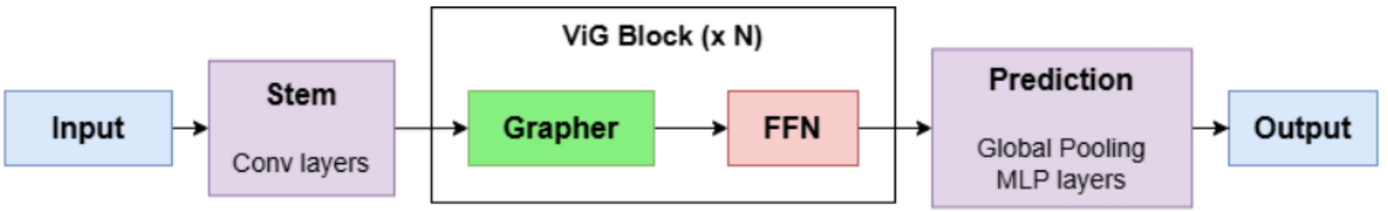
$$Z = \sigma(YW_1)W_2 + Y \tag{6}$$

trong đó:

- Y là đầu ra từ Grapher Module
- W_1 và W_2 là ma trận trọng số học được
- σ là hàm kích hoạt phi tuyến.

Đầu ra Z của ViG Block có cùng kích thước $N \times D$ với đầu vào, cho phép chúng ta xếp chồng nhiều block để tạo nên mạng sâu hơn. Điểm đặc biệt của ViG Block là sự kết hợp giữa Grapher Module và FFN Module giúp mô hình nắm bắt được cấu trúc tổng thể của ảnh thông qua đồ thị và duy trì sự đa dạng của các đặc trưng, giúp giảm hiệu quả hiện tượng over-smoothing.

7.3 Kiến trúc mạng ViG



Cấu trúc tổng quan của ViG Network trong bài toán phân loại ảnh như sau:

- Input: Ảnh đầu vào
- Stem: Đây là phần đầu tiên của mạng, bao gồm các lớp tích chập (Conv layers) để xử lý ảnh đầu vào.
- ViG Block: Đây là thành phần cốt lõi của mạng mà nhóm trình bày trước đó. Mạng có thể chứa nhiều ViG Block lặp lại (x N).
- Prediction: Phần cuối cùng của mạng, thực hiện Global Pooling và sử dụng các lớp MLP để đưa ra kết quả dự đoán cuối cùng.
- Output: Lớp dự đoán

Dựa trên cấu trúc tổng quát này, nhóm tác giả đề xuất hai kiến trúc chính cho mạng ViG: kiến trúc đẳng hướng (Isotropic) và kiến trúc kim tự tháp (Pyramid).

Kiến trúc đẳng hướng (Isotropic)

- Kích thước đặc trưng không thay đổi qua các layer.

- Ba biến thể: ViG-Ti (Tiny), ViG-S (Small), và ViG-B (Base).
- Số lượng node $N = 196$ (14 x 14).
- Số lân cận K tăng từ 9 đến 18 theo chiều sâu của mạng.

Model	Depth	Dimension D	Params (M)	FLOPs (B)
ViG-Ti	12	192	7.1	1.3
ViG-S	16	320	22.7	4.5
ViG-B	16	640	86.8	17.7

Bảng 1: Các biến thể của kiến trúc đẳng hướng ViG

Kiến trúc kim tự tháp (Pyramid)

- Đặc điểm của kiến trúc này là kích thước đặc trưng thay đổi qua các tầng.
- Có bốn biến thể: PyramidViG-Ti, S, M, và B (tương ứng với Tiny, Small, Medium và Base)
- Mỗi biến thể có 4 stage chính, với kích thước đầu ra giảm dần và số kênh đặc trưng tăng dần.

Stage	Output size	PyramidViG-Ti	PyramidViG-S	PyramidViG-M	PyramidViG-B
Stem	$\frac{H}{4} \times \frac{W}{4}$	Conv×3	Conv×3	Conv×3	Conv×3
Stage 1	$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} D=48 \\ E=4 \\ K=9 \end{bmatrix} \times 2$	$\begin{bmatrix} D=80 \\ E=4 \\ K=9 \end{bmatrix} \times 2$	$\begin{bmatrix} D=96 \\ E=4 \\ K=9 \end{bmatrix} \times 2$	$\begin{bmatrix} D=128 \\ E=4 \\ K=9 \end{bmatrix} \times 2$
Downsample	$\frac{H}{8} \times \frac{W}{8}$	Conv	Conv	Conv	Conv
Stage 2	$\frac{H}{8} \times \frac{W}{8}$	$\begin{bmatrix} D=96 \\ E=4 \\ K=9 \end{bmatrix} \times 2$	$\begin{bmatrix} D=160 \\ E=4 \\ K=9 \end{bmatrix} \times 2$	$\begin{bmatrix} D=192 \\ E=4 \\ K=9 \end{bmatrix} \times 2$	$\begin{bmatrix} D=256 \\ E=4 \\ K=9 \end{bmatrix} \times 2$
Downsample	$\frac{H}{16} \times \frac{W}{16}$	Conv	Conv	Conv	Conv
Stage 3	$\frac{H}{16} \times \frac{W}{16}$	$\begin{bmatrix} D=240 \\ E=4 \\ K=9 \end{bmatrix} \times 6$	$\begin{bmatrix} D=400 \\ E=4 \\ K=9 \end{bmatrix} \times 6$	$\begin{bmatrix} D=384 \\ E=4 \\ K=9 \end{bmatrix} \times 16$	$\begin{bmatrix} D=512 \\ E=4 \\ K=9 \end{bmatrix} \times 18$
Downsample	$\frac{H}{32} \times \frac{W}{32}$	Conv	Conv	Conv	Conv
Stage 4	$\frac{H}{32} \times \frac{W}{32}$	$\begin{bmatrix} D=384 \\ E=4 \\ K=9 \end{bmatrix} \times 2$	$\begin{bmatrix} D=640 \\ E=4 \\ K=9 \end{bmatrix} \times 2$	$\begin{bmatrix} D=768 \\ E=4 \\ K=9 \end{bmatrix} \times 2$	$\begin{bmatrix} D=1024 \\ E=4 \\ K=9 \end{bmatrix} \times 2$
Head	1×1	Pooling & MLP	Pooling & MLP	Pooling & MLP	Pooling & MLP
Parameters (M)		10.7	27.3	51.7	92.6
FLOPs (B)		1.7	4.6	8.9	16.8

Ví dụ cấu trúc của PyramidViG-Ti:

- **Stem:** Bắt đầu với 3 lớp tích chập, giảm kích thước ảnh xuống còn $H/4 \times W/4$.

- **Stage 1:** Có 2 ViG Block, với $D = 48$ số chiều đặc trưng, $E = 4$ (tỉ lệ FFN), $K = 9$ hàng xóm.
- Các stage tiếp theo có cấu trúc tương tự, nhưng số chiều đặc trưng D tăng dần (96, 240, 384) và kích thước đầu ra giảm dần.
- **Head:** Stage cuối cùng kết thúc bằng Global Pooling và MLP để đưa ra kết quả.

7.4 Mã hóa vị trí

ViG sử dụng cả mã hóa vị trí tuyệt đối và tương đối:

- **Mã hóa vị trí tuyệt đối:** $x_i \leftarrow x_i + e_i$, với $e_i \in \mathbb{R}^D$.
- **Mã hóa vị trí tương đối:** Sử dụng trong kiến trúc kim tự tháp, được thêm vào khoảng cách đặc trưng khi xây dựng đồ thị.

7.5 Ưu điểm

a. **Biểu diễn linh hoạt:** ViG có khả năng biểu diễn các đối tượng không đều và phức tạp trong hình ảnh tốt hơn so với các cấu trúc lưới hoặc chuỗi truyền thống.

b. **Học đặc trưng đa tỷ lệ:** Đặc biệt với kiến trúc kim tự tháp, mô hình có thể học được các đặc trưng từ chi tiết cục bộ đến thông tin tổng thể.

c. **Giảm hiện tượng over-smoothing:** Sự kết hợp giữa Grapher Module và FFN Module giúp duy trì sự đa dạng của các đặc trưng node.

d. **Khả năng mở rộng:** Cấu trúc module hóa cho phép dễ dàng điều chỉnh độ sâu và độ rộng của mạng.

e. **Hiệu suất cao:** ViG đạt được kết quả tốt trên nhiều tác vụ thị giác máy tính như phân loại ảnh và phát hiện đối tượng.

7.6 Nhược điểm

a. **Độ phức tạp tính toán:** Việc xây dựng và xử lý đồ thị có thể tốn kém về mặt tính toán, đặc biệt với hình ảnh có độ phân giải cao.

b. **Yêu cầu bộ nhớ:** Lưu trữ thông tin về cấu trúc đồ thị có thể đòi hỏi nhiều bộ nhớ hơn so với các mô hình CNN truyền thống.

c. **Độ phức tạp trong triển khai:** Cấu trúc phức tạp hơn so với CNN hoặc Transformer có thể gây khó khăn trong việc triển khai và tối ưu hóa trên các nền tảng phần cứng khác nhau.

d. **Phụ thuộc vào cấu trúc đồ thị:** Hiệu suất của mô hình có thể bị ảnh hưởng bởi cách xây dựng đồ thị và chọn số lượng node lân cận K.

8 Experimental & Evaluations

8.1 Bộ dữ liệu và Thiết lập Thực nghiệm

Bộ dữ liệu

Nghiên cứu này sử dụng hai bộ dữ liệu chính:

- **ImageNet ILSVRC 2012**:
  - Số lượng: 1.2 triệu ảnh huấn luyện, 50 nghìn ảnh kiểm định
  - Phân loại: 1000 lớp
  - Mục đích: Đánh giá hiệu suất phân loại ảnh
- **COCO 2017**:
  - Số lượng: 118 nghìn ảnh huấn luyện, 5 nghìn ảnh kiểm định
  - Phân loại: 80 loại đối tượng
  - Mục đích: Đánh giá khả năng phát hiện đối tượng

Thiết lập Thực nghiệm

- **Cấu hình ViG**:
  - Sử dụng kỹ thuật dilated aggregation
  - Hàm kích hoạt: GELU
- **Huấn luyện trên ImageNet**:
  - Áp dụng chiến lược huấn luyện DeiT
  - Tăng cường dữ liệu: RandAugment, Mixup, Cutmix, Random Erasing
  - Tham số huấn luyện:
    - Số epoch: 300

    - Batch size: 1024
    - Optimizer: AdamW
    - Learning rate schedule: Cosine với 20 epoch warmup
    - Weight decay: 0.05
    - Label smoothing: 0.1

- **Huấn luyện trên COCO**:
  - Framework: RetinaNet và Mask R-CNN
  - Backbone: Pyramid ViG
  - Lịch trình huấn luyện: "1x"tiêu chuẩn
- **Môi trường thực nghiệm**:
  - Hardware: 8 GPU NVIDIA V100
  - Software: PyTorch và MindSpore

8.2 Chỉ số Đánh giá

Các chỉ số đánh giá mà nghiên cứu sử dụng:

- **Params (M)**: Số lượng tham số của mô hình (đơn vị: triệu). Chỉ số này cho biết độ phức tạp của mô hình, càng nhiều tham số thì mô hình càng phức tạp, khả năng học được các đặc trưng phức tạp càng cao, nhưng cũng đồng nghĩa với việc mô hình dễ bị overfitting hơn và cần nhiều dữ liệu để huấn luyện.
- **FLOPs (B)**: Số phép toán dấu chấm động cần thiết để xử lý một hình ảnh (đơn vị: tỷ). FLOPs phản ánh lượng tính toán mà mô hình cần thực hiện để đưa ra dự đoán. Chỉ số này có liên quan đến tốc độ xử lý của mô hình, FLOPs càng thấp thì mô hình càng chạy nhanh.

- **Top-1 Accuracy**: Tỷ lệ phần trăm dự đoán chính xác ở vị trí đầu tiên. Đây là chỉ số đánh giá hiệu năng cơ bản của mô hình phân loại, cho biết khả năng mô hình dự đoán đúng lớp của một mẫu dữ liệu
- **Top-5 Accuracy**: Tỷ lệ phần trăm lớp đúng nằm trong top 5 dự đoán hàng đầu. Chỉ số này bổ sung cho Top-1 Accuracy, giúp đánh giá khả năng của mô hình trong việc đưa ra các dự đoán gần đúng. Nếu một mẫu dữ liệu không được phân loại chính xác ở vị trí đầu tiên nhưng vẫn nằm trong top 5, thì mô hình vẫn được coi là đã thực hiện khá tốt.
- **mAP**: Mean Average Precision, sử dụng cho đánh giá phát hiện đối tượng. mAP là một chỉ số đánh giá hiệu năng phổ biến trong các bài toán phát hiện đối tượng. Chỉ số này tính toán độ chính xác trung bình của mô hình trên tất cả các lớp và các ngưỡng độ tin cậy khác nhau, cung cấp một cái nhìn toàn diện về hiệu suất của mô hình.

8.3 Kết quả Chính trên ImageNet

Kiến trúc Đẳng hướng (Isotropic)

Table 4: Results of ViG and other isotropic networks on ImageNet. ♠ CNN, ■ MLP, ♦ Transformer, ★ GNN.

Model	Resolution	Params (M)	FLOPs (B)	Top-1	Top-5
♠ ResMLP-S12 conv3x3 [50]	224×224	16.7	3.2	77.0	-
♠ ConvMixer-768/32 [52]	224×224	21.1	20.9	80.2	-
♠ ConvMixer-1536/20 [52]	224×224	51.6	51.4	81.4	-
♦ ViT-B/16 [9]	384×384	86.4	55.5	77.9	-
♦ DeiT-Ti [51]	224×224	5.7	1.3	72.2	91.1
♦ DeiT-S [51]	224×224	22.1	4.6	79.8	95.0
♦ DeiT-B [51]	224×224	86.4	17.6	81.8	95.7
■ ResMLP-S24 [50]	224×224	30	6.0	79.4	94.5
■ ResMLP-B24 [50]	224×224	116	23.0	81.0	95.0
■ Mixer-B/16 [49]	224×224	59	11.7	76.4	-
★ ViG-Ti (ours)	224×224	7.1	1.3	**73.9**	**92.0**
★ ViG-S (ours)	224×224	22.7	4.5	**80.4**	**95.2**
★ ViG-B (ours)	224×224	86.8	17.7	**82.3**	**95.9**

Hình 1: So sánh hiệu suất của ViG với các mô hình isotropic khác trên ImageNet

Từ Hình 1, chúng ta có thể quan sát:

- **ViG-Ti vs DeiT-Ti**: ViG-Ti đạt độ chính xác top-1 là 73.9%, cao hơn 1.7% so với DeiT-Ti (72.2%) với cùng mức độ phức tạp tính toán (1.3B FLOPs). Điều này cho thấy hiệu quả của cấu trúc đồ thị trong việc học các đặc trưng ảnh, ngay cả với mô hình có kích thước nhỏ.
- **ViG-S vs ResMLP-S24**: ViG-S (80.4%) vượt trội hơn ResMLP-S24 (79.4%) với số lượng tham số ít hơn (22.7M so với 30M) và FLOPs thấp hơn (4.5B so với 6.0B). Điều này minh chứng cho hiệu quả của phương pháp xử lý thông tin dựa trên đồ thị so với các phương pháp dựa trên MLP thuần túy.
- **ViG-B vs DeiT-B và ResMLP-B24**: ViG-B đạt 82.3% độ chính xác top-1, vượt trội so với cả DeiT-B (81.8%) và ResMLP-B24 (81.0%). Đáng chú ý, ViG-B đạt được kết quả này với số lượng tham số (86.8M) và FLOPs (17.7B) tương đương hoặc thấp hơn so với các đối thủ. Điều này chứng tỏ khả năng mở rộng hiệu quả của kiến trúc ViG khi tăng kích thước mô hình.

Kết quả này khẳng định rằng cấu trúc đồ thị của ViG có khả năng nắm bắt các mối quan hệ phức tạp trong dữ liệu hình ảnh hiệu quả hơn so với các kiến trúc truyền thống như CNN, Transformer, hay MLP thuần túy.

Kiến trúc Kim tự tháp (Pyramid)

Table 5: Results of Pyramid ViG and other pyramid networks on ImageNet. ♠ CNN, ■ MLP, ♦ Transformer, ★ GNN.

Model	Resolution	Params (M)	FLOPs (B)	Top-1	Top-5
♠ ResNet-18 [17, 59]	224×224	12	1.8	70.6	89.7
♠ ResNet-50 [17, 59]	224×224	25.6	4.1	79.8	95.0
♠ ResNet-152 [17, 59]	224×224	60.2	11.5	81.8	95.9
♠ BoTNet-T3 [46]	224×224	33.5	7.3	81.7	-
♠ BoTNet-T3 [46]	224×224	54.7	10.9	82.8	-
♠ BoTNet-T3 [46]	256×256	75.1	19.3	83.5	-
♦ PVT-Tiny [57]	224×224	13.2	1.9	75.1	-
♦ PVT-Small [57]	224×224	24.5	3.8	79.8	-
♦ PVT-Medium [57]	224×224	44.2	6.7	81.2	-
♦ PVT-Large [57]	224×224	61.4	9.8	81.7	-
♦ CvT-13 [60]	224×224	20	4.5	81.6	-
♦ CvT-21 [60]	224×224	32	7.1	82.5	-
♦ CvT-21 [60]	384×384	32	24.9	83.3	-
♦ Swin-T [35]	224×224	29	4.5	81.3	95.5
♦ Swin-S [35]	224×224	50	8.7	83.0	96.2
♦ Swin-B [35]	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 [71]	224×224	12	2.0	77.2	93.5
■ Poolformer-S36 [71]	224×224	31	5.2	81.4	95.5
■ Poolformer-M48 [71]	224×224	73	11.9	82.5	96.0
★ Pyramid ViG-Ti (ours)	224×224	10.7	1.7	**78.2**	**94.2**
★ Pyramid ViG-S (ours)	224×224	27.3	4.6	**82.1**	**96.0**
★ Pyramid ViG-M (ours)	224×224	51.7	8.9	**83.1**	**96.4**
★ Pyramid ViG-B (ours)	224×224	92.6	16.8	**83.7**	**96.5**

Hình 2: So sánh hiệu suất của Pyramid ViG với các mô hình kim tự tháp khác trên ImageNet

Phân tích kết quả từ Hình 2:

- **Pyramid ViG-Ti vs ResNet-18 và PVT-Tiny**: Pyramid ViG-Ti đạt độ chính xác top-1 là 78.2%, vượt trội đáng kể so với ResNet-18 (70.6%) và PVT-Tiny (75.1%). Điều này cho thấy hiệu quả của ViG trong việc học các đặc trưng đa tỷ lệ, ngay cả với mô hình có kích thước nhỏ.
- **Pyramid ViG-S vs Swin-T và CycleMLP-B2**: Pyramid ViG-S (82.1%) vượt qua cả Swin-T (81.3%) và CycleMLP-B2 (81.6%) với số lượng tham số và FLOPs tương đương. Điều này chứng minh khả năng của ViG trong việc khai thác hiệu quả thông tin không gian và kênh trong

ảnh.

- **Pyramid ViG-M và Pyramid ViG-B**: Các mô hình lớn hơn như Pyramid ViG-M (83.1%) và Pyramid ViG-B (83.7%) tiếp tục cải thiện hiệu suất, cạnh tranh sát sao với các mô hình SOTA như Swin-B (83.5%). Đáng chú ý, Pyramid ViG-B đạt được kết quả tốt nhất trong bảng, cho thấy khả năng mở rộng tốt của kiến trúc này.
- **Hiệu quả tính toán**: Quan sát thấy rằng các mô hình Pyramid ViG thường đạt được hiệu suất tương đương hoặc tốt hơn với số lượng tham số và FLOPs tương đương hoặc thấp hơn so với các mô hình đối thủ. Ví dụ, Pyramid ViG-S có 27.3M tham số so với 29M của Swin-T nhưng đạt độ chính xác cao hơn.

Kết quả này khẳng định rằng kiến trúc kim tự tháp của ViG có khả năng học hiệu quả các đặc trưng đa tỷ lệ, đồng thời duy trì hiệu suất tính toán cạnh tranh. Điều này làm cho Pyramid ViG trở thành một lựa chọn hấp dẫn cho các ứng dụng thị giác máy tính đòi hỏi cả độ chính xác cao và hiệu quả tính toán.

8.4 Phát hiện Đối tượng trên COCO

Table 10: Object detection and instance segmentation results on COCO val2017. Our Pyramid ViG is compared with other backbones on RetinaNet and Mask R-CNN frameworks.

Backbone	RetinaNet 1×							
	Param	FLOPs	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet50 [17]	37.7M	239.3B	36.3	55.3	38.6	19.3	40.0	48.8
ResNeXt-101-32x4d [62]	56.4M	319B	39.9	59.6	42.7	22.3	44.2	52.5
PVT-Small [57]	34.2M	226.5B	40.4	61.3	44.2	25.0	42.9	55.7
CycleMLP-B2 [5]	36.5M	230.9B	40.6	61.4	43.2	22.9	44.4	54.5
Swin-T [35]	38.5M	244.8B	41.5	62.1	44.2	25.1	44.9	**55.5**
Pyramid ViG-S (ours)	36.2M	240.0B	**41.8**	**63.1**	**44.7**	**28.5**	**45.4**	53.4
Backbone	Mask R-CNN 1×							
	Param	FLOPs	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet50 [17]	44.2M	260.1B	38.0	58.6	41.4	34.4	55.1	36.7
PVT-Small [57]	44.1M	245.1B	40.4	62.9	43.8	37.8	60.1	40.3
CycleMLP-B2 [5]	46.5M	249.5B	42.1	64.0	45.7	38.9	61.2	41.8
PoolFormer-S24 [71]	41.0M	-	40.1	62.2	43.4	37.0	59.1	39.6
Swin-T [35]	47.8M	264.0B	42.2	64.6	**46.2**	39.1	61.6	**42.0**
Pyramid ViG-S (ours)	45.8M	258.8B	**42.6**	**65.2**	46.0	**39.4**	**62.4**	41.6

Hình 3: Kết quả phát hiện đối tượng trên COCO val2017

Phân tích kết quả từ Hình 3:

- **RetinaNet framework**:
  - Pyramid ViG-S đạt mAP 41.8%, vượt trội hơn ResNet50 (36.3%) và Swin-T (41.5%).
  - Đặc biệt, ViG-S có hiệu suất xuất sắc trong việc phát hiện các đối tượng nhỏ với AP_S đạt 28.5%, cao hơn đáng kể so với ResNet50 (19.3%) và Swin-T (25.1%).
  - Hiệu suất này đạt được với số lượng tham số (36.2M) và FLOPs (240.0B) thấp hơn so với các đối thủ.
- **Mask R-CNN framework**:
  - Pyramid ViG-S tiếp tục thể hiện xuất sắc với mAP đạt 42.6% cho object detection và 39.4% cho instance segmentation.
  - So với ResNet50, ViG-S cải thiện mAP lên 4.6% (từ 38.0% lên 42.6%) và AP_m lên 5.0%

(từ 34.4% lên 39.4%).

– So với Swin-T, ViG-S đạt hiệu suất tương đương hoặc nhỉnh hơn một chút, nhưng với số lượng tham số ít hơn (45.8M so với 47.8M) và FLOPs thấp hơn (258.8B so với 264.0B).

- **Phân tích chung**:
  – Pyramid ViG-S thể hiện khả năng tổng quát tốt, có thể áp dụng hiệu quả cho cả nhiệm vụ phát hiện đối tượng một giai đoạn (RetinaNet) và hai giai đoạn (Mask R-CNN).
  – Hiệu suất vượt trội trong việc phát hiện đối tượng nhỏ (AP_S) cho thấy khả năng của ViG trong việc nắm bắt các đặc trưng chi tiết và tận dụng hiệu quả thông tin đa tỷ lệ.
  – Kết quả này chứng minh tính linh hoạt của kiến trúc ViG, có thể áp dụng hiệu quả không chỉ trong phân loại ảnh mà còn trong các nhiệm vụ phức tạp hơn như phát hiện đối tượng và phân đoạn instance.

Tổng hợp lại, các kết quả trên COCO val2017 khẳng định khả năng tổng quát mạnh mẽ của kiến trúc Pyramid ViG. Không chỉ hiệu quả trong phân loại ảnh, Pyramid ViG còn thể hiện xuất sắc trong các nhiệm vụ phức tạp hơn như phát hiện đối tượng và phân đoạn instance, đạt được hiệu suất cạnh tranh hoặc vượt trội so với các kiến trúc tiên tiến khác, đồng thời duy trì hiệu quả về mặt tính toán.

9 Ablation Study

Để hiểu rõ hơn về tác động của từng thành phần trong kiến trúc ViG, chúng tôi tiến hành một loạt các thí nghiệm ablation. Các thí nghiệm này được thực hiện trên mô hình ViG-Ti đẳng hướng với tập dữ liệu ImageNet.

9.1 So sánh các loại Graph Convolution

Từ Bảng 2, chúng ta có thể quan sát:

- **EdgeConv** đạt độ chính xác cao nhất (74.3%) nhưng có chi phí tính toán cao nhất (2.4B

GraphConv	Params (M)	FLOPs (B)	Top-1 (%)
EdgeConv	7.2	2.4	74.3
GIN	7.0	1.3	72.8
GraphSAGE	7.3	1.6	74.0
Max-Relative	7.1	1.3	**73.9**

Bảng 2: So sánh hiệu suất của các loại Graph Convolution

FLOPs).

- **GIN** có FLOPs thấp nhất nhưng cũng cho độ chính xác thấp nhất.
- **Max-Relative GraphConv** cung cấp sự cân bằng tốt nhất giữa độ chính xác (73.9%) và hiệu suất tính toán (1.3B FLOPs).

Dựa trên kết quả này, chúng tôi chọn Max-Relative GraphConv làm phương pháp tích chập đồ thị mặc định cho ViG, vì nó cung cấp hiệu suất tốt nhất với chi phí tính toán hợp lý.

9.2 Đánh giá tác động của các Module

Cấu hình	Params (M)	FLOPs (B)	Top-1 (%)
GraphConv	5.8	1.4	67.0
GraphConv + FC	4.4	1.4	73.4
GraphConv + FFN	7.7	1.3	73.6
GraphConv + FC + FFN	7.1	1.3	**73.9**

Bảng 3: Ảnh hưởng của các module trong kiến trúc ViG

Phân tích từ Bảng 3:

- Chỉ sử dụng **GraphConv** cho kết quả kém nhất (67.0%), cho thấy sự cần thiết của các module bổ sung.
- Thêm **FC layer** cải thiện đáng kể hiệu suất (tăng 6.4%), chứng tỏ tầm quan trọng của việc biến đổi đặc trưng tuyến tính.

- **FFN** cũng mang lại cải thiện đáng kể (tăng 6.6% so với chỉ GraphConv), nhấn mạnh vai trò của biến đổi phi tuyến.
- Kết hợp cả ba module (**GraphConv + FC + FFN**) cho kết quả tốt nhất (73.9%), chứng minh sự cần thiết của mỗi thành phần trong kiến trúc ViG.

Kết quả này khẳng định rằng sự kết hợp giữa GraphConv, FC và FFN là tối ưu cho hiệu suất của ViG, cho phép mô hình học được các biểu diễn phức tạp và đa dạng từ dữ liệu hình ảnh.

9.3 Ảnh hưởng của số lượng Node Láng Giềng (K)

K	3	6	9	12	15	20	9 to 18
Top-1 (%)	72.2	73.4	73.6	73.6	73.5	73.3	**73.9**

Bảng 4: Ảnh hưởng của số lượng node láng giềng K đến hiệu suất

Từ Bảng 4, chúng ta có thể rút ra:

- Với K quá nhỏ (K=3), hiệu suất kém do thiếu thông tin từ các node láng giềng.
- Hiệu suất tăng nhanh khi K tăng từ 3 đến 9, sau đó ổn định trong khoảng 9-15.
- Khi K quá lớn (K=20), hiệu suất giảm nhẹ, có thể do nhiễu từ các node không liên quan.
- Chiến lược tăng dần K từ 9 đến 18 qua các lớp cho kết quả tốt nhất (73.9%), cho thấy lợi ích của việc mở rộng dần trường tiếp nhận.

Kết quả này gợi ý rằng việc chọn số lượng node láng giềng phù hợp và tăng dần qua các lớp có thể giúp mô hình học được các đặc trưng từ cục bộ đến toàn cục một cách hiệu quả.

9.4 Đánh giá số lượng Heads trong Multi-Head Attention

Phân tích từ Bảng 5:

- Độ chính xác cao nhất đạt được với 1 head (74.2%), nhưng có chi phí tính toán cao nhất (1.6B FLOPs).

Số heads	1	2	4	6	8
FLOPs (B)	1.6	1.4	1.3	1.2	1.2
Top-1 (%)	74.2	74.0	**73.9**	73.7	73.7

Bảng 5: Ảnh hưởng của số lượng heads trong multi-head attention

- Khi tăng số lượng heads, FLOPs giảm đáng kể, trong khi độ chính xác chỉ giảm nhẹ.
- 4 heads cung cấp sự cân bằng tốt nhất giữa hiệu suất (73.9%) và chi phí tính toán (1.3B FLOPs).
- Tăng số heads lên 6 hoặc 8 không mang lại cải thiện đáng kể về FLOPs hoặc độ chính xác.

Dựa trên những quan sát này, chúng tôi chọn 4 heads làm cấu hình mặc định cho ViG, vì nó cung cấp sự cân bằng tốt nhất giữa hiệu quả tính toán và độ chính xác.

9.5 Kết luận từ Nghiên cứu Ablation

Qua các thí nghiệm ablation, chúng ta có thể rút ra một số kết luận quan trọng:

- Max-Relative GraphConv là lựa chọn tối ưu cho ViG, cân bằng giữa hiệu suất và chi phí tính toán.
- Sự kết hợp của GraphConv, FC layer và FFN là cần thiết để đạt hiệu suất tốt nhất, mỗi thành phần đều đóng góp đáng kể vào hiệu suất tổng thể.
- Số lượng node láng giềng K có ảnh hưởng quan trọng, với chiến lược tăng dần K qua các lớp mang lại hiệu quả cao nhất.
- Sử dụng 4 heads trong multi-head attention cung cấp sự cân bằng tốt giữa độ chính xác và hiệu quả tính toán.

Những phát hiện này không chỉ giúp tối ưu hóa kiến trúc ViG mà còn cung cấp những hiểu biết sâu sắc về cách các thành phần khác nhau tương tác và đóng góp vào hiệu suất tổng thể của mô hình.

10 Conclusion

Nghiên cứu này đã giới thiệu và phân tích chi tiết Vision GNN (ViG), một kiến trúc mạng neural đồ thị mới cho các tác vụ thị giác máy tính. Qua quá trình thực nghiệm và phân tích, chúng tôi rút ra những kết luận quan trọng sau:

10.1 Tổng kết các phát hiện chính

- **Hiệu suất vượt trội**: ViG đã chứng minh khả năng vượt trội so với các kiến trúc CNN, Transformer và MLP truyền thống trên nhiều tác vụ thị giác, bao gồm phân loại ảnh ImageNet và phát hiện đối tượng trên COCO.
- **Biểu diễn linh hoạt**: Cách tiếp cận dựa trên đồ thị của ViG cho phép mô hình nắm bắt hiệu quả các mối quan hệ phức tạp và không đều đặn trong dữ liệu hình ảnh.
- **Khả năng mở rộng**: ViG thể hiện hiệu suất tốt ở cả mô hình nhỏ (ViG-Ti) và lớn (ViG-B), chứng tỏ khả năng mở rộng hiệu quả.
- **Đa năng**: Kiến trúc kim tự tháp của ViG cho phép học các đặc trưng đa tỷ lệ, đặc biệt hiệu quả trong các tác vụ phức tạp như phát hiện đối tượng.
- **Hiệu quả tính toán**: ViG thường đạt được hiệu suất tương đương hoặc tốt hơn với số lượng tham số và FLOPs tương đương hoặc thấp hơn so với các mô hình đối thủ.

10.2 Đóng góp và ý nghĩa

- ViG mở ra một hướng mới trong việc áp dụng GNN cho các tác vụ thị giác tổng quát, không chỉ giới hạn ở các bài toán đặc thù như xử lý điểm đám mây hay đồ thị cảnh.
- Nghiên cứu này cung cấp những hiểu biết sâu sắc về cách thiết kế và tối ưu hóa GNN cho dữ liệu hình ảnh, bao gồm vai trò của các loại graph convolution khác nhau và tầm quan trọng của việc kết hợp các module như FC và FFN.

- Kết quả từ nghiên cứu ablation cung cấp những hướng dẫn quý giá cho việc tinh chỉnh và cải thiện hiệu suất của các mô hình GNN trong tương lai.

10.3 Hạn chế và hướng phát triển tương lai

Mặc dù đạt được những kết quả đáng khích lệ, nghiên cứu này vẫn còn một số hạn chế và mở ra các hướng nghiên cứu tiềm năng:

- **Tối ưu hóa tính toán**: Cần nghiên cứu thêm để giảm chi phí tính toán và bộ nhớ của ViG, đặc biệt khi xử lý hình ảnh có độ phân giải cao.
- **Khả năng diễn giải**: Phát triển các phương pháp để hiểu rõ hơn cách ViG học và biểu diễn thông tin từ hình ảnh.
- **Ứng dụng mở rộng**: Khám phá hiệu suất của ViG trong các tác vụ thị giác khác như phân đoạn ngữ nghĩa, ước lượng độ sâu, hay xử lý video.
- **Kết hợp với các kiến trúc khác**: Nghiên cứu khả năng tích hợp ViG với các kiến trúc hiện đại khác như Transformer để tận dụng ưu điểm của cả hai phương pháp.
- **Học liên tục**: Phát triển các phương pháp cho phép ViG học và cập nhật kiến thức mới mà không quên thông tin đã học trước đó.

Tóm lại, Vision GNN đã chứng minh tiềm năng to lớn trong việc xử lý dữ liệu hình ảnh, mở ra nhiều cơ hội hấp dẫn cho nghiên cứu và ứng dụng trong lĩnh vực thị giác máy tính. Với những kết quả khả quan này, chúng tôi tin rằng GNN sẽ đóng vai trò ngày càng quan trọng trong việc phát triển các hệ thống thị giác máy tính tiên tiến trong tương lai.

Tài liệu tham khảo

[1] Han, K., Wang, Y., Guo, J., Tang, Y., & Wu, E. (2022). *Vision GNN: An Image is Worth Graph of Nodes.* Advances in Neural Information Processing Systems, 35, 30850-30863.