Graph Mining Seminar

Vision HGNN: An Image is More than a Graph of Nodes

Thành viên

- Đoàn Ngọc Mai - 21127104
- Lê Nguyễn Kiều Oanh - 21127129
- Dương Trường Bình - 21127229
- Lê Phước Quang Huy - 21127616

Giảng viên

- Lê Ngọc Thành
- Lê Nhựt Nam

Ho Chi Minh City, August 2024

Content

01 Overview

02 Method

03 Experimental & Evaluations

04 Conclusion

01

Overview

01 Tổng quan

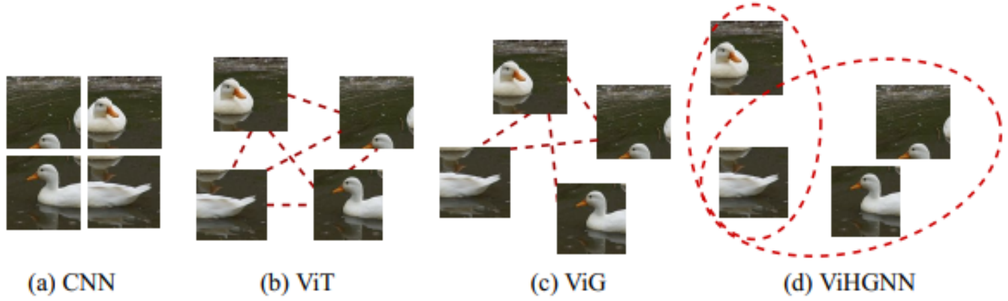
02 Khoảng trống nghiên cứu

03 Công trình dưa trên

01

Tổng quan

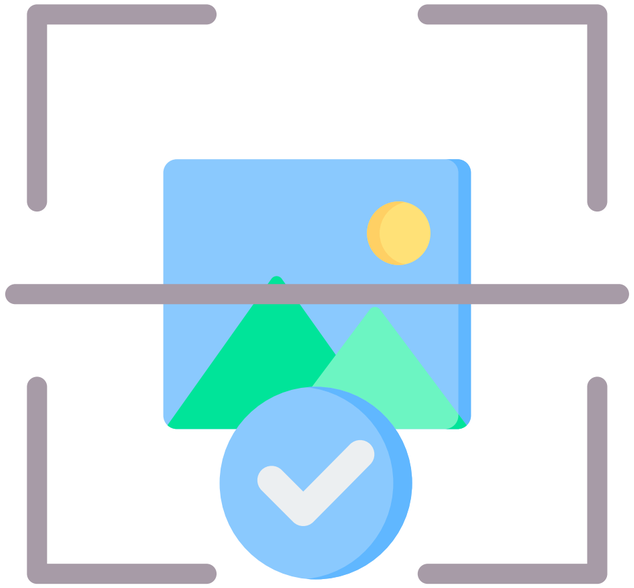
- *Hypergraph Neural Networks (HGNN) là một mô hình mạng nơ-ron sử dụng* ***cấu trúc siêu đồ thị*** *để biểu diễn và xử lý dữ liệu.*
- *HGNN* ***vượt trội hơn*** *các mô hình GNN truyền thống bằng cách khai thác tốt hơn các tương tác bậc cao trong dữ liệu.*



01

Khoảng trống nghiên cứu

- **Gap:** Chưa có mô hình sử dụng **siêu đồ thị** để biểu diễn hình ảnh trong thị giác máy tính



01

Công trình dựa trên

Nguồn cảm hứng:

- Công trình không phải là một hướng hoàn toàn mới mà là một sự phát triển dựa trên công trình trước đó về **Vision GNN**, giải quyết những hạn chế của nó bằng cách sử dụng siêu đồ thị để biểu diễn hình ảnh.

02

Method

01 Ký hiệu và Định nghĩa

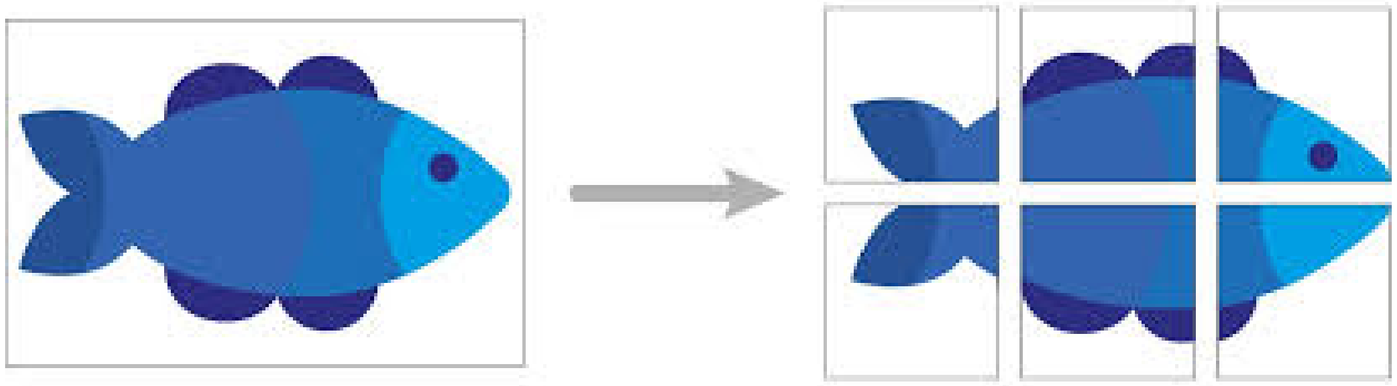
02 Định nghĩa về siêu đồ thị

03 Hypergraph Neural Network

04 ViHGNN Framework

02

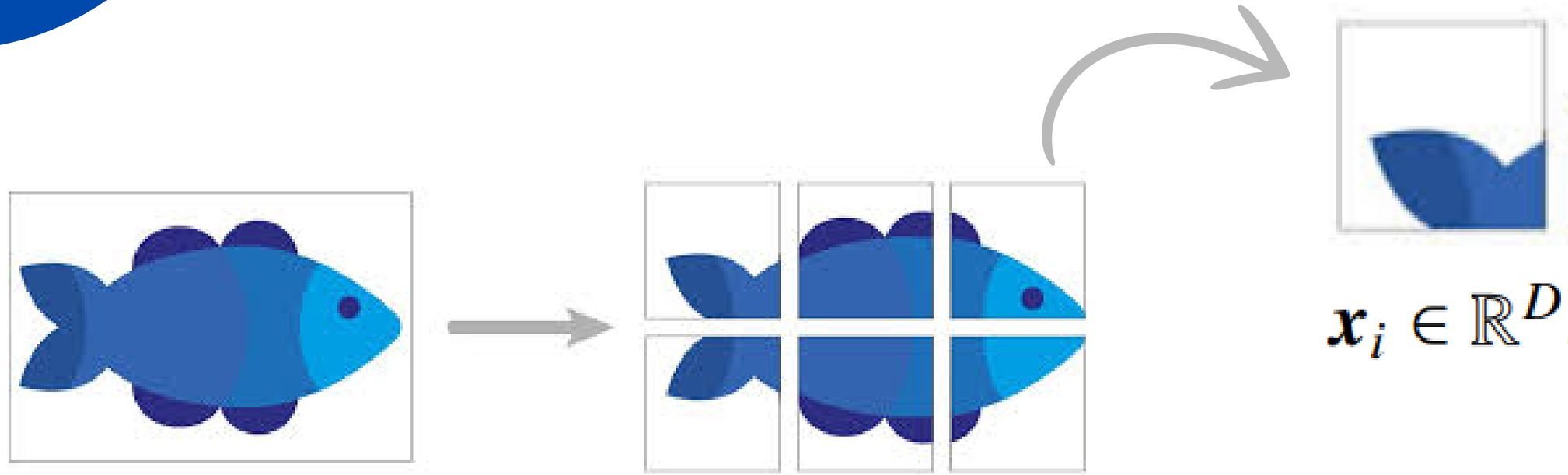
Ký hiệu và Định nghĩa



$$I \in R^{H \times W \times 3}$$

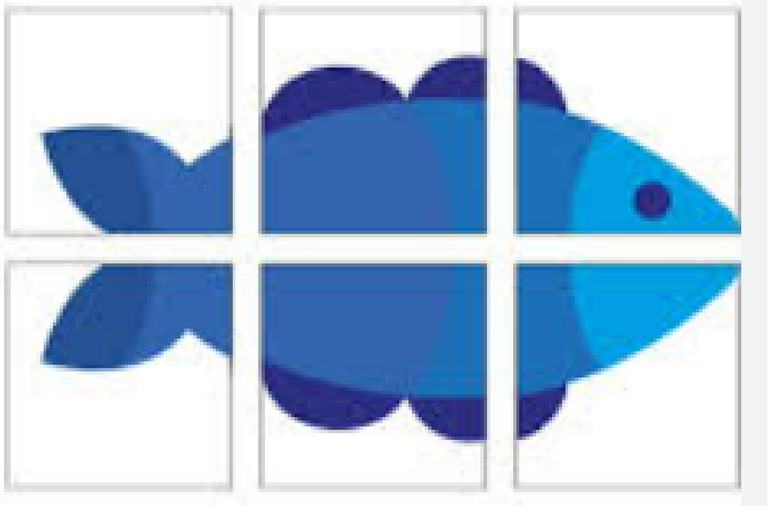
02

Ký hiệu và Định nghĩa



02

Ký hiệu và Định nghĩa



$$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$$

02

Định nghĩa về siêu đồ thị

$G = (V, E)$

V: tập hợp N đỉnh riêng biệt

02

Định nghĩa về siêu đồ thị

$G = (V, E)$

V: tập hợp N đỉnh riêng biệt

E: tập hợp các siêu cạnh

02

Định nghĩa về siêu đồ thị

Siêu cạnh được biểu diễn như sau:

$$\boldsymbol{H} \in \{0, 1\}^{N \times E}$$

$H_{ie} = 1$ nếu siêu cạnh e∈E chứa đỉnh vi∈V

$H_{ie} = 0$ nếu ngược lại

02

Định nghĩa về siêu đồ thị

Với mỗi đỉnh và siêu cạnh, bậc của chúng được biểu diễn như sau

$$D_{ii} = \sum_{e=1}^{E} H_{ie}$$

$$B_{ee} = \sum_{i=1}^{N} = H_{ie}$$

02

Hypergraph Neurol Network

Công thức tổng quát của lớp tích chập siêu đồ thị:

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

- W: ma trận trọng số của các siêu cạnh
- Θ1 và Θ2: lần lượt là các tham số có thể học được của lớp HGNN
- σ là một hàm kích hoạt
- X^l và X^(l+1): các node embeddings (nhúng nút) đầu vào và đầu ra.

02 Hypergraph Neurol Network

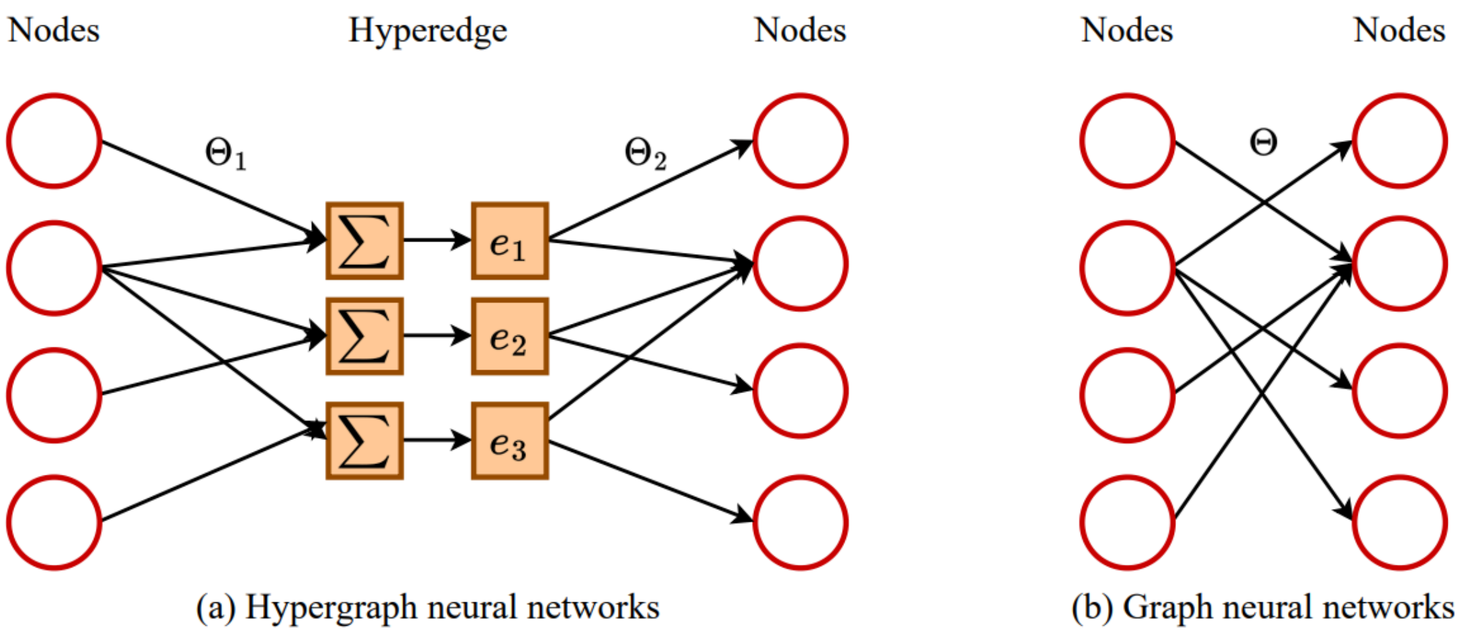


Figure 3. The comparison of message passing mechanisms in (a) HGNNs and (b) GNNs.

02

Framework

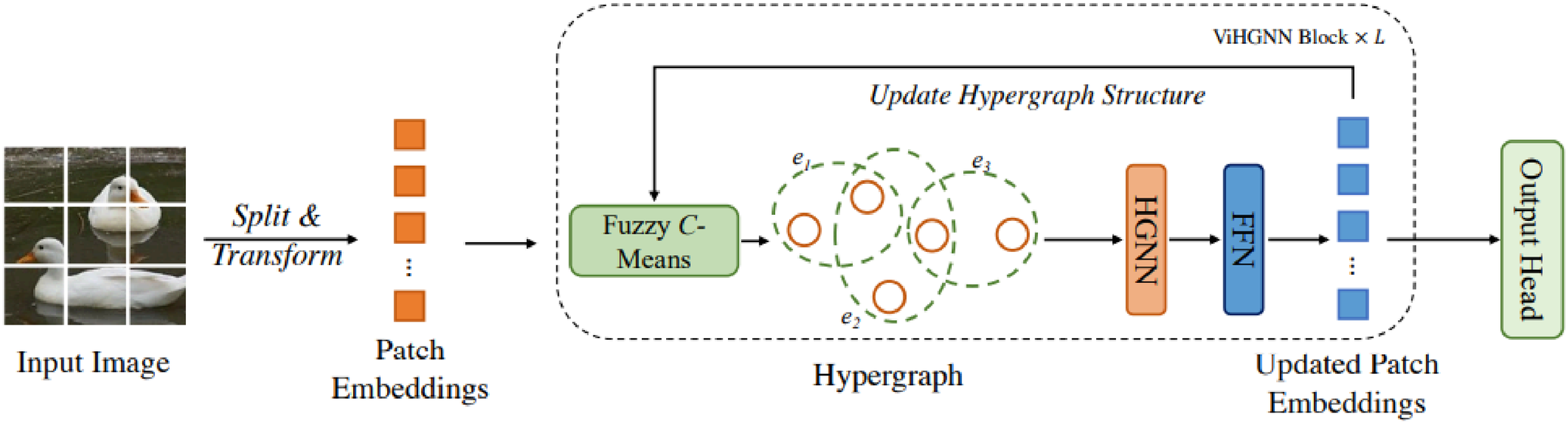


Figure 2. Overview of our ViHGNN model. HGNN and FNN represent hypergraph neural network and feed-forward network, respectively.

02 Framework

Kết luận

- **Pyramid ViHGNN** có xu hướng **phức tạp hơn và mạnh mẽ hơn** so với **Isotropic ViHGNN**, thể hiện qua số lượng tham số và FLOPs lớn hơn. Tuy nhiên, điều này cũng đòi hỏi tài nguyên tính toán nhiều hơn.
- **Isotropic ViHGNN** có cấu trúc **đơn giản** hơn, phù hợp cho các ứng dụng yêu cầu hiệu suất tính toán thấp, trong khi **Pyramid ViHGNN** thích hợp cho các tác vụ đòi hỏi khả năng học cao hơn từ dữ liệu hình ảnh.

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

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	1×1	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

03

Experimental & Evaluations

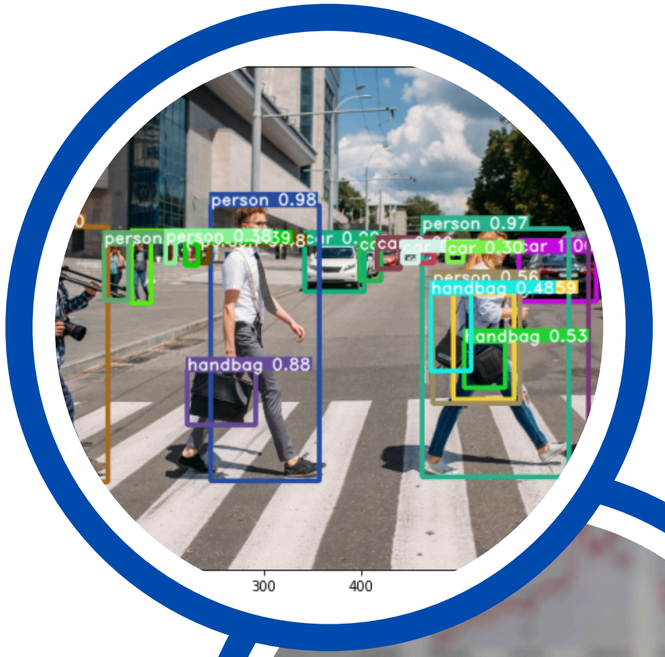
01 Quy trình thực hiện

02 Kết quả thực nghiệm

03

Quy trình thực hiện

- Áp dụng ViHGNN trên nhiệm vụ phân loại ảnh
- Thử nghiệm để đánh giá ảnh hưởng của từng thành phần trong mô hình ViHGNN
- Áp dụng trên nhiệm vụ phát hiện đối tượng
- Sử dụng **PyTorch** cho việc triển khai và **8 GPU NVIDIA V100** trên phiên bản AWS EC2 để huấn luyện mô hình



Experimental Settings

Datasets:

- **ImageNet ILSVRC 2012** cho nhiệm vụ phân loại ảnh
- **COCO 2017** cho nhiệm vụ phát hiện đối tượng

Baselines:

- **Isotropic:** Isotropic ViHGNN sẽ được so sánh với các mô hình đẳng hướng khác như ResMLP, ConvMixer, ViT, DeiT
- **Pyramid:** Pyramid ViHGNN sẽ được so sánh với các mô hình như ResNet, BoTNet, PVT, CVT, Swin Transformer, CycleMLP, Poolformer.

Hyper-parameters Settings:

- **Object Detection:** dùng RetinaNet và Mask R-CNN làm framework chung.
- **Image Classification:** dùng các chiến lược huấn luyện phổ biến trong DeiT trên ImageNet.

Table 3. Training hyper-parameters for ImageNet.

ViHGNN	Ti	S	M	B
Epochs	300			
Optimizer	AdamW [44]			
Batch size	1024			
Start learning rate (LR)	2e-3			
Learning rate schedule	Cosine			
Warmup epochs	20			
Weight decay	0.05			
Label smoothing [52]	0.1			
Stochastic path [26]	0.1	0.1	0.1	0.3
Repeated augment [24]	✓			
RandAugment [8]	✓			
Mixup prob. [76]	0.8			
Cutmix prob. [75]	1.0			
Random erasing prob. [82]	0.25			
Exponential moving average	0.99996			

03

Image Classification

Kiến trúc Isotropic

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**

Kiến trúc Pyramid

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**

Ablation Study

Các thử nghiệm để đánh giá **ảnh hưởng của từng thành phần** trong mô hình ViHGNN:

- Phương pháp xây dựng siêu đồ thị
- Hiệu suất của các phương pháp phân cụm khác nhau
- Số vòng lặp cập nhật
- Số lượng siêu cạnh
- Chi phí của việc học cấu trúc siêu đồ thị

Ablation Study

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** cho thấy Fuzzy C-Means mang lại hiệu suất tốt nhất với chi phí thấp.

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

Ablation Study

Các phương pháp phân cụm khác nhau:

- So sánh DBSCAN, Mean Shift Clustering, Spectral Clustering và Fuzzy C-Means
- Kết quả cho thấy Fuzzy C-Means cho kết quả tốt nhất

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**

Ablation Study

Số vòng lặp cập nhật:

- Thử nghiệm với 1, 2 và 3 vòng lặp cập nhật cho thấy không có sự khác biệt đáng kể về độ chính xác.

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

Ablation Study

Số lượng siêu cạnh:

- Việc tăng số lượng siêu cạnh không đảm bảo hiệu suất tốt hơn và có thể dẫn đến chi phí tính toán cao hơn.
- Việc phân bổ số lượng siêu cạnh cho các tầng nên tương ứng với kích thước đặc trưng của từng tầng

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

Ablation Study

Chi phí của việc học cấu trúc siêu đồ thị:

- So sánh thời gian chạy trung bình (mili giây) trên mỗi mẫu của ViT-Ti và Isotropic ViHGNN-Ti để đánh giá chi phí
- **Kết quả:** 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

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

Object Detection

- Đánh giá khả năng **tổng quát hóa** của ViHGNN bằng cách sử dụng Pyramid ViHGNN-S làm bộ xương chính cho RetinaNet và Mask R-CNN

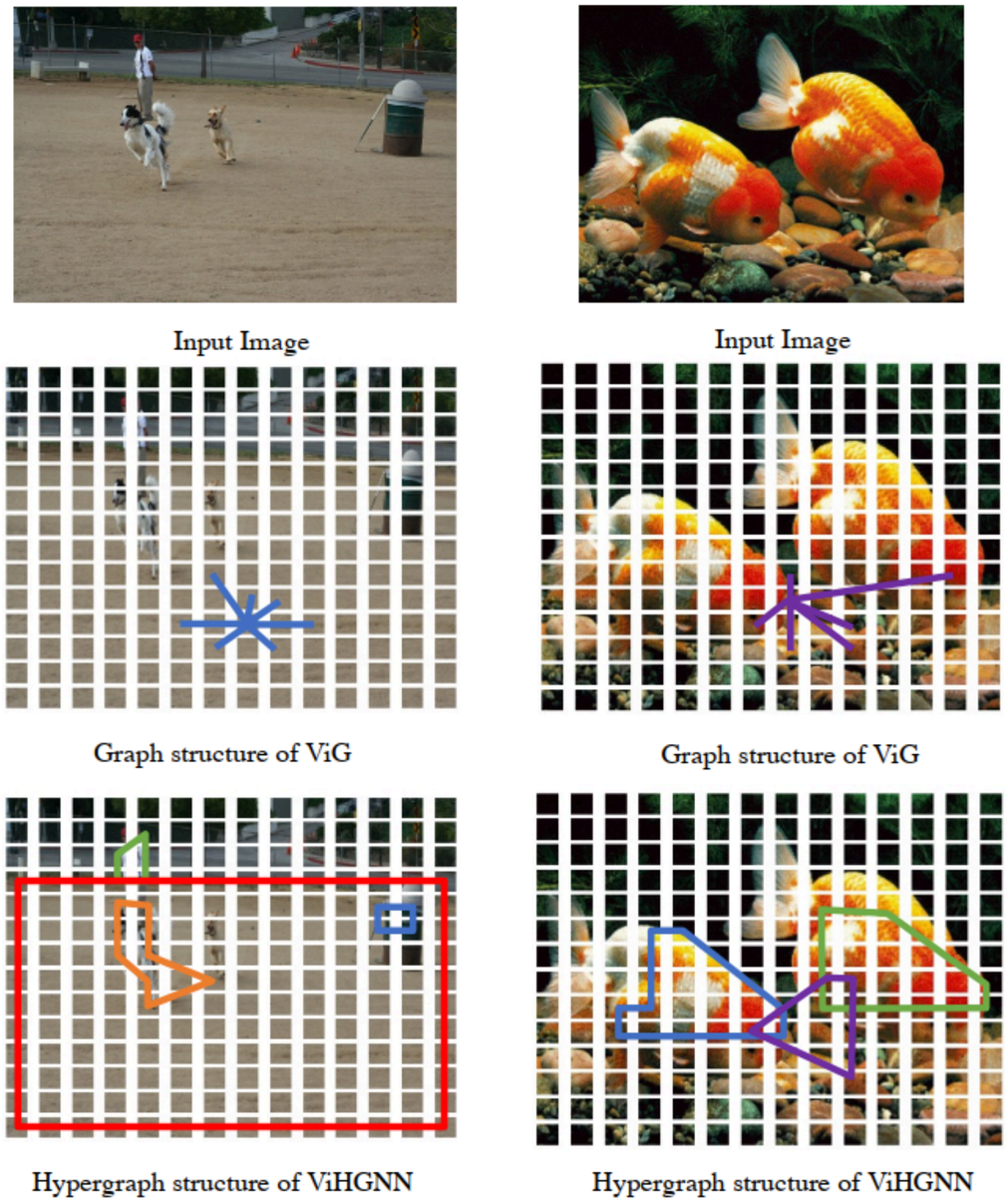
- **Kết quả:** ViHGNN vượt trội hơn các bộ xương chính khác như ResNet3, CycleMLP4 và Swin Transformer5.

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**

ViG vs ViHGNN

- ViG tạo ra các **cạnh dư thừa**
- ViHGNN nắm bắt các **mối quan hệ bậc cao** hơn và giảm thiểu kết nối nhiễu



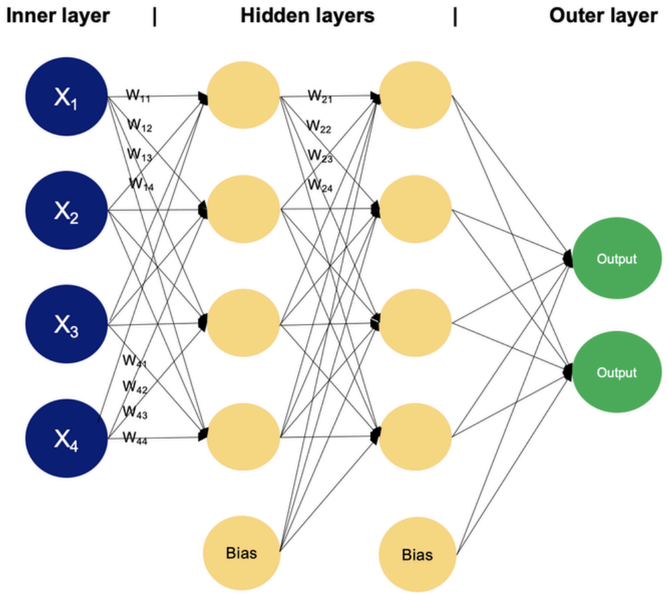
Conclusion

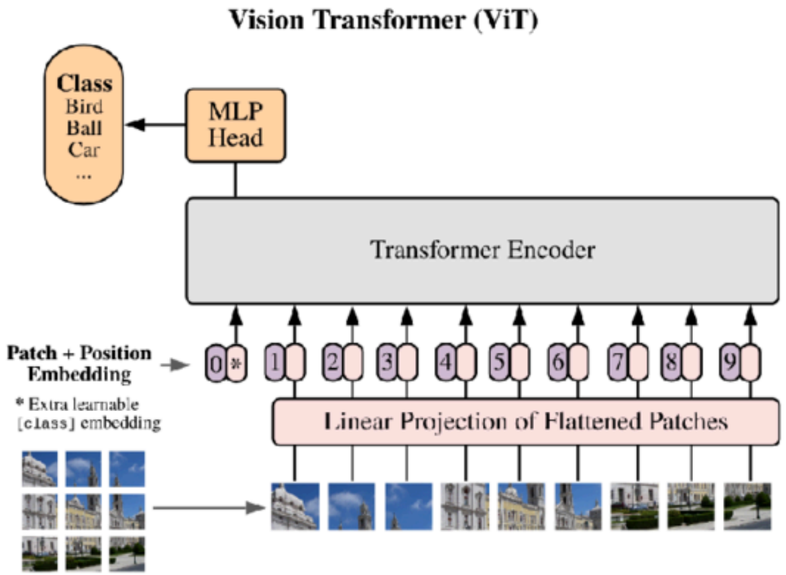
- **Đổi mới**: Giới thiệu ViHGNN, một mô hình mới sử dụng siêu đồ thị để biểu diễn hình ảnh. Khác với các phương pháp trước đây như ViG chỉ sử dụng đồ thị đơn giản, ViHGNN tận dụng khả năng của siêu đồ thị trong việc nắm bắt các mối quan hệ phức tạp bậc cao giữa các mảng hình ảnh.
- **Thách thức**: Việc xác định số lượng siêu cạnh tối ưu là một thách thức và xây dựng cấu trúc siêu đồ thị tối ưu cho một hình ảnh là một bài toán khó.
- **Kết quả**: ViHGNN đạt được hiệu suất vượt trội so với các mô hình mạng khác, bao gồm cả ViG, CNN, ViT và MLP, trên cả hai nhiệm vụ phân loại ảnh ImageNet và phát hiện đối tượng COCO.
- **Tác động**: ViHGNN mở ra một hướng đi mới cho việc biểu diễn hình ảnh bằng siêu đồ thị, cho thấy tiềm năng ứng dụng của siêu đồ thị trong các tác vụ thị giác máy tính.

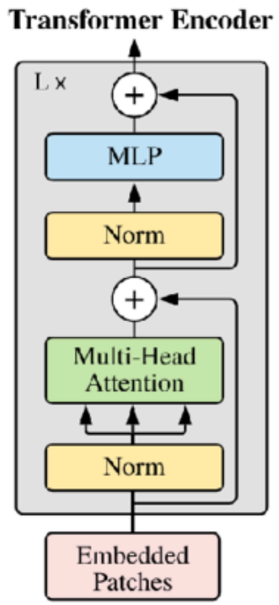
Thank You

01

Bối cảnh


Inner layer
Hidden layers
Outer layer
X1
X2
X3
X4
W11
W12
W13
W14
W21
W22
W23
W24
W31
W42
W43
W44
Output
Output
Bias
Bias

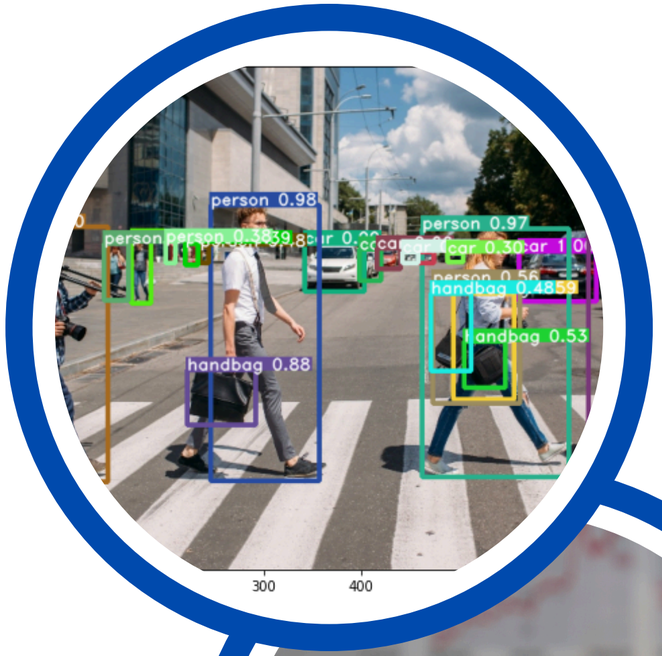

Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
Linear Projection of Flattened Patches


Transformer Encoder
L x
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches

02

Ý Nghĩa Thực Tế và Ứng Dụng

- Ứng dụng trong phân loại và phát hiện đối tượng
- Tối ưu hóa các hệ thống thị giác trong thời gian thực



03

Problem Statements

01 Các mạng thị giác truyền thống (như **CNN, ViT**) gặp khó khăn trong việc nắm bắt đầy đủ các mối quan hệ phức tạp giữa các phần tử trong ảnh

02 **Vision GNN** không biểu diễn được mối quan hệ cao và tạo ra nhiều cạnh thừa

03 **Vision HGNN** nắm bắt được mối quan hệ phức tạp, tối ưu hóa hiệu quả tính toán và hiệu suất vượt trội

01

Bối cảnh

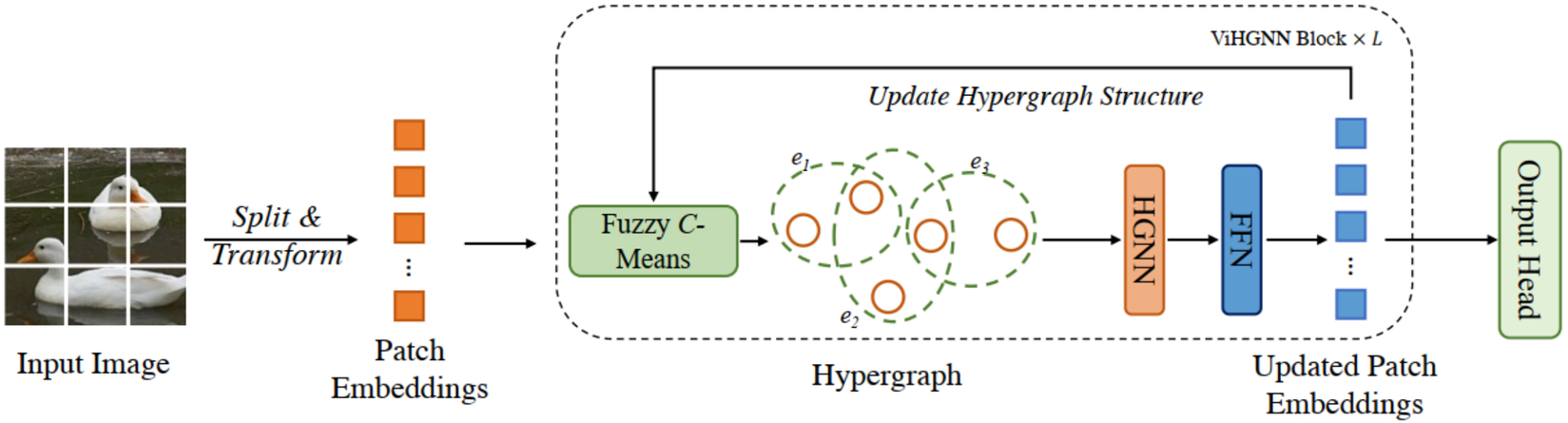


Figure 2. Overview of our ViHGNN model. HGNN and FNN represent hypergraph neural network and feed-forward network, respectively.

02

Ý Nghĩa Khoa Học

- Đóng góp vào lý thuyết về mạng neural đồ thị
- Tiến bộ trong học cấu trúc đồ thị

02

Động lực

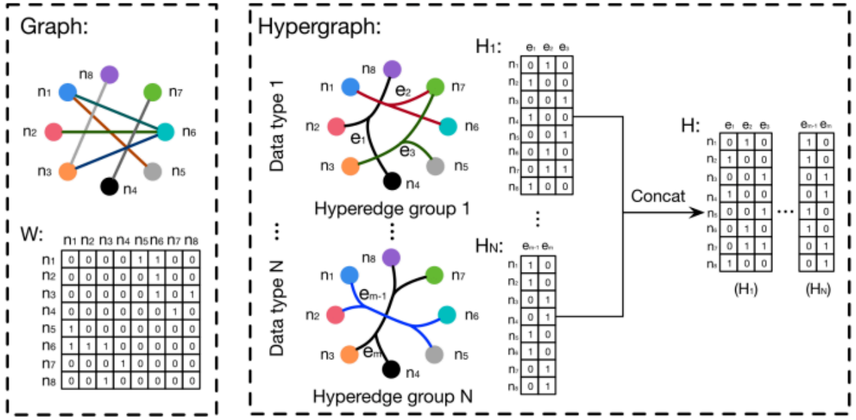
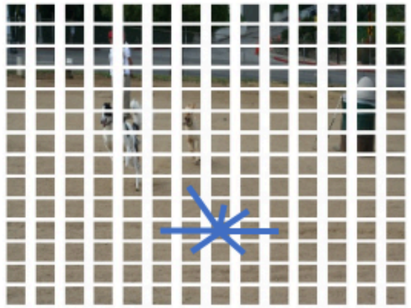
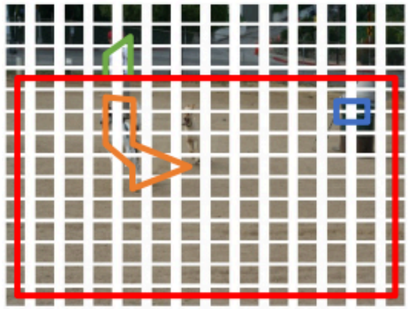


Figure 2: The comparison between graph and hypergraph.

Input Image



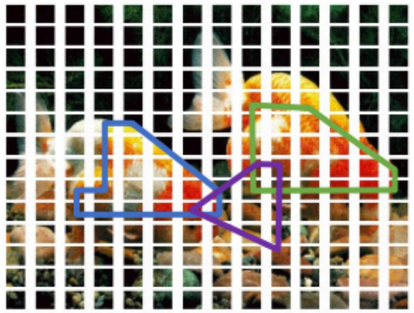
Graph structure of ViG



Hypergraph structure of ViHGNN

Input Image

Graph structure of ViG



Hypergraph structure of ViHGNN

04

Related Work

01 Timeline

02 Khoảng trống nghiên cứu

03 Công trình dựa trên

04

RELATED WORK

1
2012
CNN

2
2014
VGGNet

3
2015
- ResNet
- Inception

4
2016

5
2017
- GNN
- Transformer

6
2018
HGNN

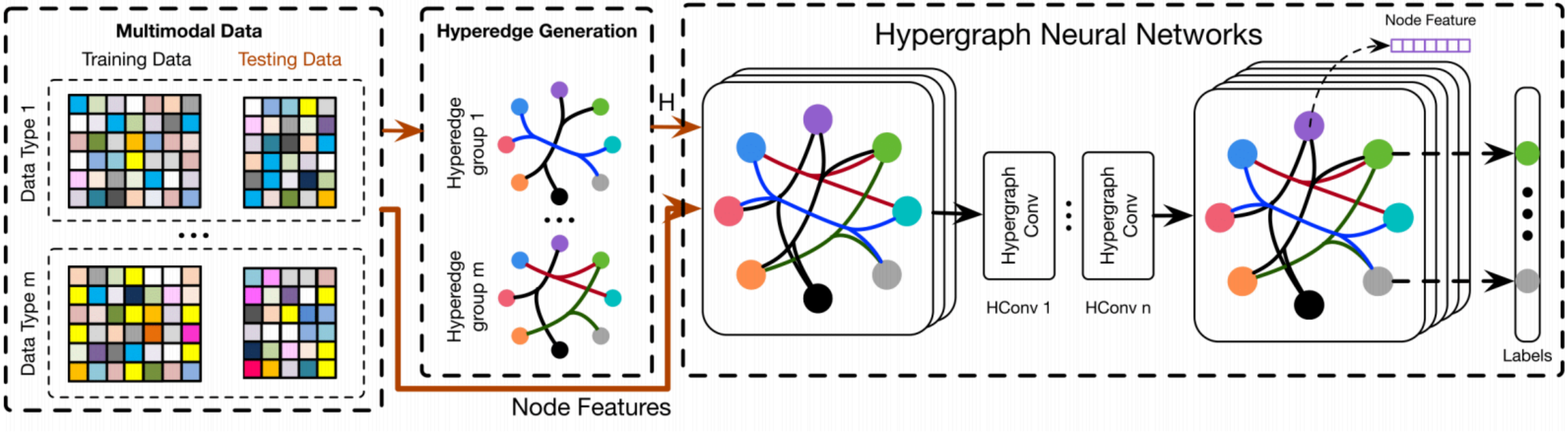
7
2020
ViT

8
2021
- Các biến thể ViT
- MLP

9
2022
- **Vision GNN**
- DHSL

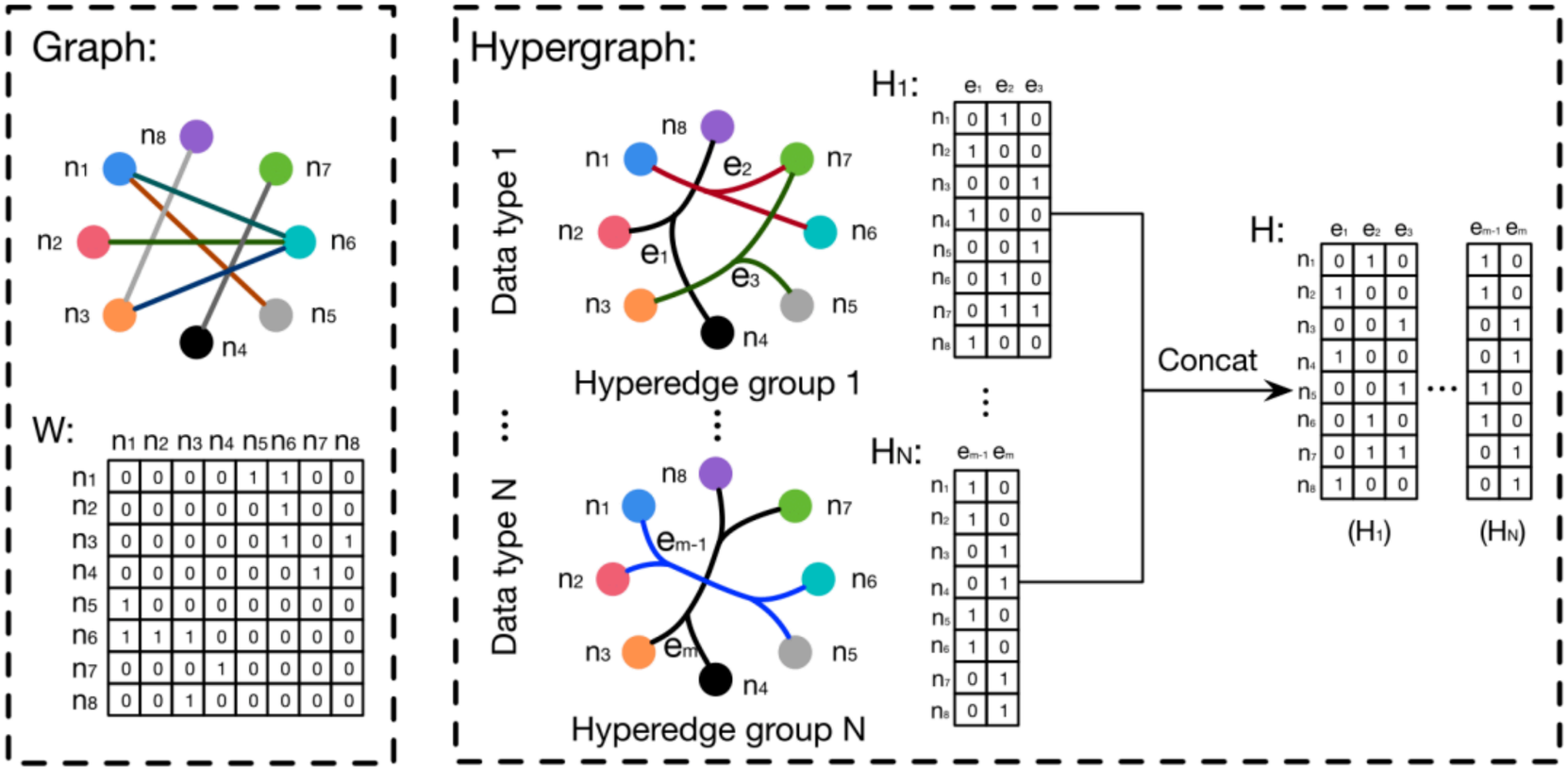
10
2023
Vision HGNN

HGNN


Multimodal Data
Training Data
Testing Data
Data Type 1
Data Type m
Hyperedge Generation
Hyperedge group 1
Hyperedge group m
H
Node Features
Hypergraph Neural Networks
Node Feature
Hypergraph Conv
Hypergraph Conv
HConv 1
HConv n
Labels

04

Graph vs Hypergraph



04

RELATED WORK

1
2012
CNN

2
2014
VGGNet

3
2015
- ResNet
- Inception

4
2016

5
2017
- GNN
- Transformer

6
2018
HGNN

7
2020
ViT

8
2021
- Các biến thể ViT
- MLP

9
2022
- **Vision GNN**
- DHSL

10
2023
Vision HGNN

05 Plan

CÔNG VIỆC	NGHIÊN CỨU VỀ PHƯƠNG PHÁP VÀ KẾT QUẢ THỰC NGHIỆM	TÌM HIỂU SOURCE CODE	CHẠY THỰC NGHIỆM	PHÂN TÍCH SO SÁNH KẾT QUẢ VÀ BÁO CÁO	ABLATION STUDY (NẾU CÒN THỜI GIAN)
MÔ TẢ	• Đọc và hiểu rõ các thành phần của Vision GNN và kết quả thực nghiệm • Tìm hiểu và chuẩn bị các bộ dữ liệu cần thiết: ImageNet và COCO.	• Tìm kiếm, đọc và hiểu rõ source code có liên quan. • Đảm bảo rằng các phần mềm và công cụ cần thiết đã được cài đặt và hoạt động đúng cách.	Chạy các thí nghiệm để so sánh ViG với các nhóm kiến trúc như CNN, MLP, Transformer	• Tổng hợp kết quả thực nghiệm và viết báo cáo. • Chuẩn bị slide và thuyết trình	Thực hiện các nghiên cứu ablation để phân tích sâu hơn về các yếu tố ảnh hưởng đến hiệu suất của Vision GNN.
THỜI GIAN (ƯỚC TÍNH)	• 1 tuần	• 1 tuần	• 1 tuần	• 1 tuần	• 1 tuần