1. Thu thập dữ liệu – Dataset Collection

Mỗi người mang theo thiết bị điện thoại, thiết bị này sẽ liên tục phát gói tin probe gồm MAC address, RSSI,Thời gian. Sau đó họ dùng 14 thiết bị Raspberry Pi 3B+ để thu, thu được những thông tin đó, nhưng chỉ dùng RSS

Bài báo chỉ sử dụng RSS(Received Signal Strength), không dùng MAC, vì:

Ở mục I.Introduction: However, the challenges from persons with multi ple WiFi-enabled mobile devices, uncertain sniffing, and MAC address randomization [12] severely affect this kind of method and make it only get a rough accuracy of less than 80%. Search bài báo là có.

WiFi sensing – Raspberry Pi hoạt động như WiFi sniffers:

* Hệ thống sử dụng 14 thiết bị Raspberry Pi 3B+ để thu dữ liệu WiFi từ thiết bị di động của người đi qua (điện thoại, laptop...).
* Mỗi Raspberry được cấu hình như một WiFi sniffer, tức:

Không phát sóng, chỉ nghe lén các gói tin beacon/probe được phát ra bởi thiết bị di động.

Ghi lại các thông tin như: MAC address, RSS (cường độ tín hiệu nhận được), timestamp, channel, v.v.

Tác giả tự thu thập mẫu vị trí sẵn, đại loại là tạo **fingerprint database**:

1.Sử dụng 6 điện thoại

2.Di chuyển trên 300 cho mỗi thiết bị. Cách di chuyển là sliding window 60s, bước mỗi 1s.

*Sliding window là kỹ thuật chia nhỏ một chuỗi thờ gian dài thành các đoạn ngắn có độ dài cố định, các đoạn này gắn vào nhau, di chuyển từng chút, từng chút một theo thời gian.* ***Khi một thiết bị di chuyển, mỗi giây sẽ có nhiều gói probe được thu lại. thay vì dùng riêng từng gói, tác giả gom data thành các đoạn thời gian dài 60s, rồi tạo FINGERPRINT từ đó.*** Mỗi đoạn trượt đi 1 giây -> tạo thành nhiều fingerprint nối vào nhau.

Ví dụ đề bài cho 300 giây:

|  |  |  |
| --- | --- | --- |
| FIngerprint | Khoảng thời gian | Mục đích |
| 1 | Từ giây 0 -59s | Lấy RSSI trong này tạo finger đầu tiên |
| 2 | 1 - 60 | Trượt thêm 1s, tạo finger 2 |
| 3 | 2 - 61 | … |
| … | … | … |
| 241 | Từ giây 240 – 299 | Fingerprint cuối |

Tổng thu được 241 fingerprint từ 300s di chuyển, thực tế trong bài có nhắc là **more than 300s.**

Vì sao phải lấy 60s + 1s bước?

1.RSSI ko ổn định -> nếu lấy tách rời, dễ nhiễu

2.60s em nghĩ là ổn, vì tác giả cũng nghĩ vậy

3.Trượt 1s để lấy mẫu kín, ko có kẽ hở. -> KO BỎ SÓT BẤT KÌ CHUYỂN ĐỘNG NÀO

Mỗi fingerprint gồm một toạ độ được tác giả SET sẵn:

Ví dụ: Thầy đứng ở chỗ này, nó có toạ độ theo dataset (x,y) (15,50). Là trong fingerprint này chứa giá trị RSSI của 14 thiết bị thu: (x=10,y=15)⇒[r1​,r2​,...,r14​]

Với 6 thiết bị thu, mỗi thiết bị di chuyển hơn 300 điểm -> Tổng cộng hàng ngàn fingerprint.

Lí do vì sao dùng 6 thiết bị? Vì dùng 1 cái dễ overfitting.

Sau đó, dữ liệu RSS từ nhiều sniffer được sử dụng để ước lượng vị trí thiết bị thông qua thuật toán KNN (k = 3) dựa trên fingerprint WiFi.

Fingerprint: Được xây dựng trước đó bằng cách quét tín hiệu WiFi tại từng vị trí đã biết (offline phase) → tạo thành một "bản đồ mẫu" – model map.

Pha 2: người thật đi vào khu vực: sniffer thu được tín hiệu -> tạo ra RSSI, sau đó tính khoảng cách bằng Euclidean giữa tín hiệu mới này với cái tín hiệu ông nhân viên đã quay trước trong fingerprint.

Sau đó knn hoạt động, chọn k mẫu gần nhất (ở đây là 3) r tính trung bình toạ độ

Gán toạ độ đó là vị trí cái điện thoại, tức là vị trị người

Em tự rút ra lưu ý: 1.Lỡ họ mang theo nhiều thiết bị, ta có thể xử lý bằng cái MAC address để tránh bị đếm trùng.

Cách KNN hoạt động: Với mỗi thiết bị phát hiện tại thời điểm t:

* So sánh **tập RSSI quan sát được** với toàn bộ các mẫu fingerprint trong database.
* Tìm **k vị trí gần nhất** (ở đây là k = 3) theo khoảng cách Euclidean giữa vector RSSI.
* Các điểm mà knn thu được tạo thành toạ độ (x,y) trong data fingerprint -> mỗi hình người là một (x,y)

**VIDEO:**

* 5 camera smartphone được đồng bộ thời gian, gắn cố định tại các vị trí bao phủ toàn bộ khu vực (~4000 m²). Tại độ cao 2.1m
* Ghi lại video liên tục trong 2280 giây. – 38 phút
* Mỗi video được chia thành 1 frame/s và được gán nhãn thủ công số lượng người có mặt trong từng khung hình

2. Tiền xử lý dữ liệu – Data Preprocessing

1.Wifi – chuyển thành ảnh WDM

2.Video – chuyển thành ảnh ROI

- WiFi – tạo WDM (WiFi Density Map):

Vị trí của các thiết bị (đã định vị bằng KNN) được đặt vào ảnh lưới 3d – kích thước 140x80. Mỗi

DÙng KNN = 3 để định vị:

(x,y) = 1/3 của ba cái toạ độ k gần nhất

Ảnh WDM được làm mượt bằng ***Gaussian kernel*** (σ = 2) để phản ánh mật độ người thay vì chỉ đếm điểm rời rạc.

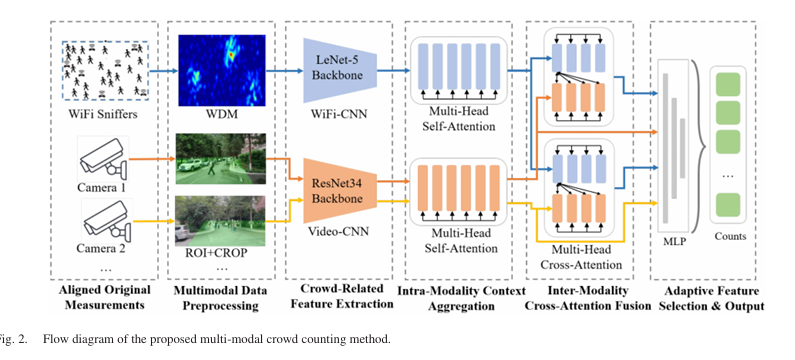
Kích thước: 140x80

TỔNG QUÁT: MỖI NGƯỜI – 1 ĐIỂM TOẠ ĐỘ TRÊN HÌNH

-Video: Cắt vùng ROI (Region of Interest) – Nghĩa là loại bỏ background, chỉ lấy vùng có người

Sau đó crop và resize về 80×160 nhằm giảm nhiễu và đồng nhất dữ liệu.

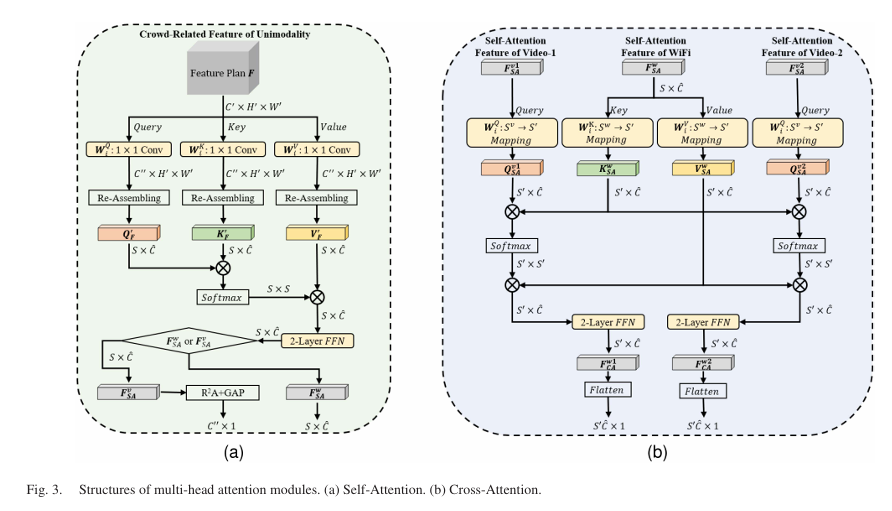
-Tương thích với CNN (Cụ thể ở đây là Resnet34)



3.Cơ chế hoạt động của hệ thống HDANet

Đoạn này dịch trong phần B. The proposed multi-modal crowd counting method

* **Trích xuất, tối ưu hóa và kết hợp đặc trưng đa phương thức dựa trên HDANet**:  
  Thứ nhất, theo mức độ trừu tượng của các đặc trưng liên quan đến đám đông trong dữ liệu đã được tiền xử lý của hai phương thức, các mạng CNN với kiến trúc khác nhau được sử dụng để trích xuất các đặc trưng đám đông tương ứng.  
  Thứ hai, để khai thác đầy đủ mối tương quan ngữ cảnh nội phương thức, cơ chế **self-attention** được sử dụng để tối ưu hóa các đặc trưng trong từng phương thức.  
  Thứ ba, cơ chế **cross-attention** và mạng nơ-ron truyền thẳng (FFN) được sử dụng để khớp và khái quát hóa các mối quan hệ giữa các phương thức.  
  Cuối cùng, một mạng perceptron nhiều lớp (MLP) được sử dụng để chọn lọc thích nghi các đặc trưng self-attention và cross-attention, và đưa ra đầu ra đếm đám đông ở phạm vi cục bộ và/hoặc toàn cục.



* 1. Self Attention

Học mqh giữa các vị trí không gian trong cùng một MODAL (Video học chính nó, Wifi học chính nó)

Các công thức sau đây dùng cho Wifi-wifi, Video-video, mạnh thằng nào thằng nấy học.

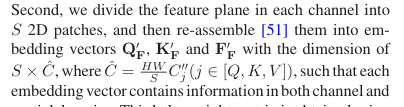
Nhìn hình, đầu tiên là Feature Plan F: Đây là đầu ra của backbone CNN

Tiếp theo là ba thành phần Query, Key, Value



Mỗi thành phần được tính từ đặc trưng đầu vào F, thông qua các lớp conv 1x1

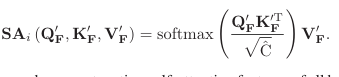
Re-Assembling:



Các đặc trưng này đc biến đổi từ tensor không gian thành ma trận 2D.

Dạng này sẽ reshape ảnh lại.

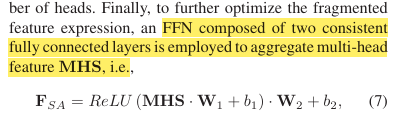
Bước 3: Tính Attention map:



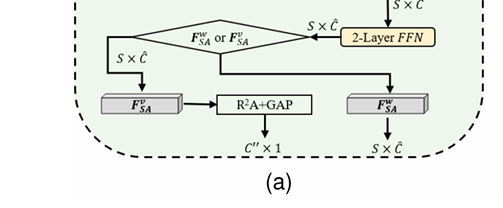
Ma trận này đại loại là cho biết ở vị trí nào, thì nên tập trung vào vị trí nào khác trong ảnh.

* QF′,KF′,VF′Q'\_F, K'\_F, V'\_FQF′​,KF′​,VF′​: là các vector đặc trưng sau khi "Re-Assembling", tức đã flatten từ tensor không gian C^×H×W về ma trận S×C^, với S=H⋅W
* C^\_sqrt: là hệ số chuẩn hoá giúp ổn định gradient (theo nguyên gốc của Transformer)
* Công thức này chính là **Scaled Dot-Product Attention** dạng chuẩn, dùng trong cả Self-Attention và Cross-Attention (khi áp dụng khác nguồn dữ liệu)

Sau đó kết hợp với Value để có output:



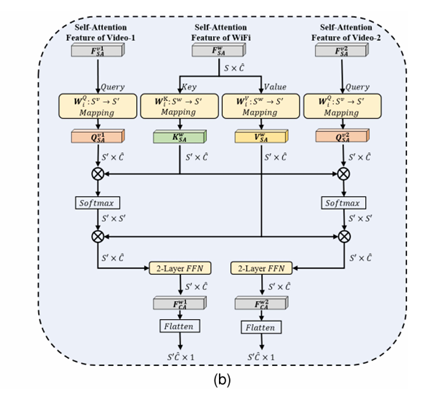
* Sau khi tính ra attention, output được đưa qua MLP.



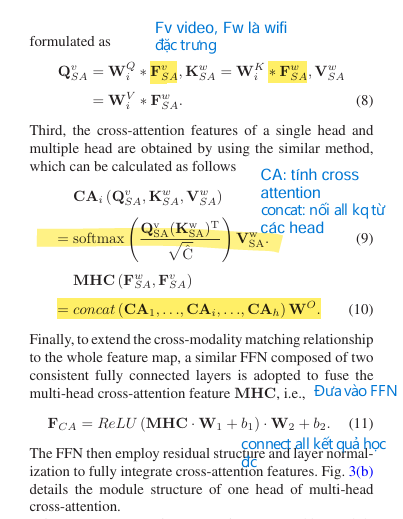
Nếu là video: thêm R^2.A + GAP để giảm về còn 1 chiều

Wifi: Giữ lại toàn bộ

* 1. Cross Attention



Video làm query, wifi làm key và value



(9) là scaled dot product attention của hai model

(11) FCA là output cross-attention sau khi đi qua FFN

DỊCH PHẦN CUỐI CÙNG, CÔNG THỨC MLP

