TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN HỌC

ỨNG DỤNG XỬ LÝ ẢNH SỐ VÀ VIDEO SỐ

Đề tài: **Ước Lượng Mật Độ Đám Đông**

Giáo viên hướng dẫn: PGS.TS. Lý Quốc Ngọc
TS. Nguyễn Mạnh Hùng

Sinh viên thực hiện:

Nguyễn Huỳnh Xuân Mai	1712091
Nguyễn Anh Khoa	1712532
Huỳnh lê Minh Nhật	1712632
Võ Văn Quân	1712698

Tp. Hồ Chí Minh, tháng 08/2020

MỤC LỤC

1.ĐỘNG LỰC NGHIÊN CỨU

1.1. ĐẶT VẤN ĐỀ

- Ngày nay, công nghệ không ngừng phát triển – đặc biệt là sự phát triển mạnh mẽ và không ngừng của AI (Artificial Intelligence) – đã ảnh hưởng rất nhiều đến đời sống xã hội. Những công nghệ này đã giải quyết nhiều bài toán thực tế trong đời sống và là công cụ đô thị hóa.
- Khi có người yêu cầu chúng ta xem ở một khu vực hiện đang có bao nhiêu người/xe thì việc đơn giản nhất là chúng ta sẽ đếm. Nhưng nó chỉ tốt khi mật độ nhỏ (vì sự **che khuất** lẫn nhau giữa các đối tượng, **kích thước** đa dạng của các đối tượng, sự **biến dạng** từ góc nhìn, quá ít **pixel** biểu diễn đối tượng . . .), đối với những mật độ lớn (như ảnh bên dưới) thì ta phải làm như thế nào. Đó là bài toán thực tế để đáp ứng nhu cầu.
- Những thiết bị công nghệ như camera giám sát làm thành một hệ thống giám sát theo dõi tình hình an ninh ở khu vực (sân bay, trường học, concert, sân vận động, …) và các cơ quan (thuộc về cá nhân hoặc chính quyền). Hay thiết thực nhất và cũng là chủ thể mà nhóm em muốn tìm hiểu đó là:
    - Hệ thống giám sát và ước lượng mật độ giao thông (xe cộ) ở những khu vực tắc nghẽn giao thông, hoặc ở những con đường ở những khung thời gian cao điểm.
    - Hệ thống giám sát và ước lượng mật độ đám đông/tụ tập của người.




- Việc duy trì sự tập trung của con người cho việc giám sát một cách liên tục dễ gây quá tải cho người giám sát, bên cạnh đó đòi hỏi chi phí vận hành, quản lý cao và khó giám sát

1.2. ĐỘNG LỰC VỀ MẶT KHOA HỌC

- Với tính ứng dụng cao và thiết thực, nhưng đây là một bài toán vẫn còn tồn tại rất nhiều thách thức và khó khăn cần phải giải quyết và cải thiện như: sự che khuất lẫn nhau giữa các đối tượng, kích thước đa dạng của các đối tượng (tùy thuộc vào độ sâu của ảnh), sự biến dạng đối tượng từ góc nhìn, quá ít pixel để biểu diễn cho đối tượng... vv. Từ đó, thúc đẩy nghiên cứu để tìm ra những hướng tiếp cận và giải pháp mới có thể khắc phục những thách thức trên, đảm bảo độ chính xác cũng như tốc độ cao nhất.

1.3. ĐỘNG LỰC VỀ MẶT ỨNG DỤNG

- Ước lượng đám đông:
    - Nghiên cứu về hành vi của con người là một chủ đề rất được quan tâm của khoa học và có lẽ là một nguồn nghiên cứu vô tận. Một trong những chủ đề nghiên cứu được trích dẫn và phổ biến nhất trong phân tích hành vi của con người là nghiên cứu về các đặc điểm và đặc điểm của đám đông.
    - Trong những năm gần đây, phân tích đám đông đã thu hút được nhiều sự quan tâm chủ yếu nhờ vào nhiều ứng dụng như giám sát an toàn, quản lý thảm họa, thiết kế không gian công cộng và thu thập thông tin tình báo, đặc biệt là trong các cảnh tắc nghẽn như đấu trường, trung tâm mua sắm và sân bay.
    - Đếm đám đông, nội địa hóa và ước tính mật độ là những mục tiêu quan trọng của một hệ thống phân tích đám đông tự động.
    - Kiến thức chính xác về quy mô đám đông, vị trí và mật độ trong không gian công cộng có thể cung cấp cái nhìn sâu sắc có giá trị cho các nhiệm vụ như quy hoạch thành phố, phân tích mô hình mua sắm của người tiêu dùng cũng như duy trì an toàn đám đông nói chung.
    - Một số nghiên cứu cố gắng đưa ra một ước tính chính xác về số lượng người thực sự có mặt trong một cảnh đông đúc thông qua ước tính mật độ.
    - Tuy nhiên, đếm đám đông và ước tính mật độ không phải là một nhiệm vụ tầm thường. Một số thách thức chính như sự xuất hiện nghiêm trọng, độ chiếu sáng kém, phối cảnh camera và môi trường rất năng động làm phức tạp thêm phân tích đám đông. Công việc phát hiện khuôn mặt trong đám đông rất phức tạp vì hiển thị khuôn mặt con người khác nhau bao gồm màu sắc, tư thế, biểu cảm, vị trí, định hướng và chiếu sáng. Hơn nữa, chất lượng kém của dữ liệu chú thích làm tăng sự phức tạp của việc đếm đám đông và phân tích hành vi trong môi trường đông đúc.
    - Các bộ dữ liệu điểm chuẩn ước tính mật độ và số lượng đám đông hiện tại không chỉ bị giới hạn về số lượng, mà còn thiếu về chiến lược chú thích
- Ước lượng xe cộ:
    - Trong thế giới hiện đại, các trung tâm đô thị đang phát triển với tốc độ rất cao. Phát triển với họ là tắc nghẽn giao thông đường bộ. Ùn tắc giao thông, đặc biệt là vào giờ cao điểm, đã trở thành thói quen. Do đó, quản lý giao thông là một trong những vấn đề quan trọng nhất ở các thị trấn hiện nay.
    - Các thành phố đang phát triển cần thống kê sử dụng xe hơi để lên kế hoạch nâng cấp cơ sở hạ tầng. Kiểm soát hiệu quả có thể đạt được khi sử dụng dữ liệu thời gian thực. Một cách tiếp cận thị giác máy tính có chi phí cài đặt thấp hơn và tính linh hoạt cao hơn so với các vòng cảm ứng. Các luồng video có sẵn công khai làm cho giải pháp này hấp dẫn.

- Một số lựa chọn thay thế đang được tìm kiếm để đối phó với vấn đề. Chúng bao gồm: mở rộng mạng lưới đường bộ, điều chỉnh số lượng phương tiện trên đường và triển khai Hệ thống giao thông thông minh (ITS). Ước tính mật độ giao thông đường bộ cung cấp thông tin quan trọng trong Hệ thống giao thông thông minh (ITS) để lập kế hoạch đường bộ, định tuyến đường thông minh, kiểm soát giao thông đường bộ, lập lịch giao thông mạng, định tuyến và phổ biến. Tính toán chính xác mật độ lưu lượng là rất cần thiết để phát triển hệ thống cảnh báo sớm và báo hiệu tự động, thống kê, lập kế hoạch và một số ứng dụng bảo mật. Hơn nữa, dữ liệu mật độ có thể được sử dụng để giúp người lái xe chọn cách tối ưu giữa nhiều tuyến đường khác nhau.
- Khác với các ITS, các lựa chọn thay thế khác (tuy nhiên hiệu quả) có nhiều thách thức thực tế trong việc thực hiện. Các ITS dựa trên một loạt các công nghệ như cảm biến vòng lặp và hệ thống giám sát video. Các ITS dựa trên tầm nhìn đã tỏ ra thuận lợi so với các phương pháp truyền thống dựa trên các cảm biến vòng lặp. Trong các hệ thống hiện đại này, camera giám sát video được lắp đặt dọc theo các con đường và ngã tư đường, nơi chúng được sử dụng để thu thập dữ liệu giao thông. Dữ liệu sau đó được phân tích để có được các thông số giao thông như mật độ giao thông đường bộ. Một cách tiếp cận đơn giản và phổ biến để ước tính mật độ giao thông đường bộ vào ban ngày bằng cách sử dụng thuật toán xử lý hình ảnh và thị giác máy tính được trình bày. Nó có thể cung cấp thông tin hình ảnh chất lượng cao một cách hiệu quả và ổn định. Thật dễ dàng và kinh tế để cài đặt máy quay video. Bên cạnh đó, nó sẽ không bao giờ làm hỏng đường, cũng sẽ không cản trở giao thông.
- Với sự phát triển nhanh chóng của thị giác máy tính và công nghệ xử lý hình ảnh kỹ thuật số, hệ thống phát hiện lưu lượng truy cập dựa trên video đã trở nên ngày càng mạnh mẽ, thời gian thực và thông minh. Dữ liệu video được thu thập trước tiên được chia thành các khung sau đó được xử lý trước theo một loạt các bước. Cuối cùng, các phương tiện được phát hiện và trích xuất từ các hình ảnh và được tính. Sau đó, mật độ giao thông được lấy là số lượng phương tiện trên một đơn vị diện tích của phần đường.
- Cố gắng giải quyết vấn đề phát hiện và đếm xe trong các cảnh giao thông tự nhiên bằng cách sử dụng các hệ thống giám sát video cho cả cảnh giao thông tự do và di chuyển chậm hoặc dừng. Cảnh giao thông đứng yên hoặc di chuyển chậm có ít báo cáo về chúng và phần lớn các hệ thống được đề xuất sử dụng các phương pháp tiếp cận dựa trên phát hiện chuyển động và do đó không phù hợp cho những cảnh này. Điều này bất chấp thực tế là việc di chuyển chậm hoặc giao thông cố định là vấn đề chính mà các cơ quan quản lý giao thông phải đối mặt ở hầu hết các thị trấn trên thế giới.

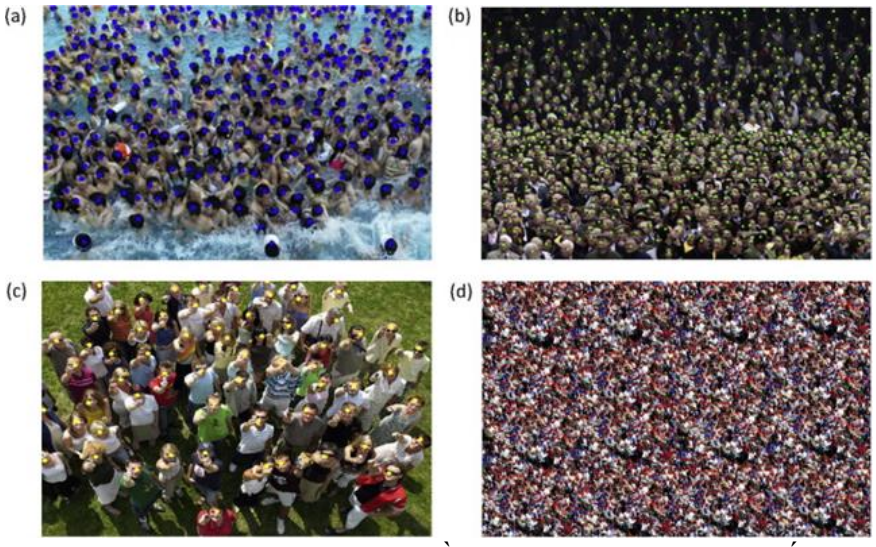
2.BỐI CẢNH TRONG & NGOÀI NƯỚC

Bài toán ước lượng đám đông đã được giải quyết từ lâu và ngày càng được nâng cấp về độ chính xác và tốc độ.

2.1. ƯỚC LƯỢNG ĐÁM ĐÔNG

a. NGOÀI NƯỚC

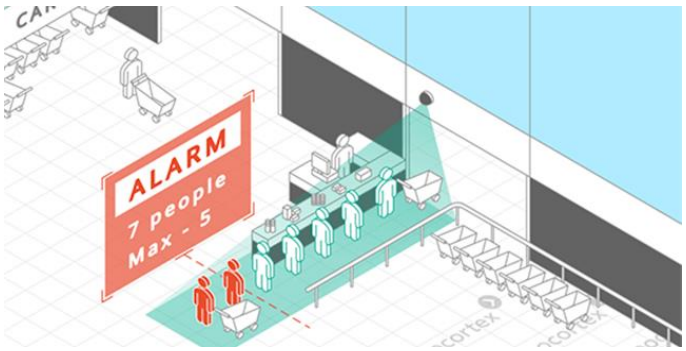
- Trước đây các hệ thống chỉ xử lý chính xác những input có mật độ thấp nhưng với những nghiên cứu cải cách gần đây hệ thống đã có thể xử lý được những input có mật độ ngày càng lớn



- Các hệ thống ước lượng đám đông đã góp phần không nhỏ trong việc thống kê tương đối số lượng người tham gia các sự kiện. Như số lượng khán giả đến xem một trận bóng đá (vào cửa tự do), Số lượng người tham gia hòa nhạc/concert, số người trên phố đi bộ, số lượng công nhân đi làm mỗi ngày, …

b. TRONG NƯỚC

Ở Việt Nam hiện nay các hệ thống ước lượng số người ở các khu vực công cộng vẫn chưa phổ biến, đa phần hiện tại là các hệ thống đếm số học sinh ở trường học, hệ thống đếm số người ra vào cửa hàng, trung tâm thương mại, siêu thị.

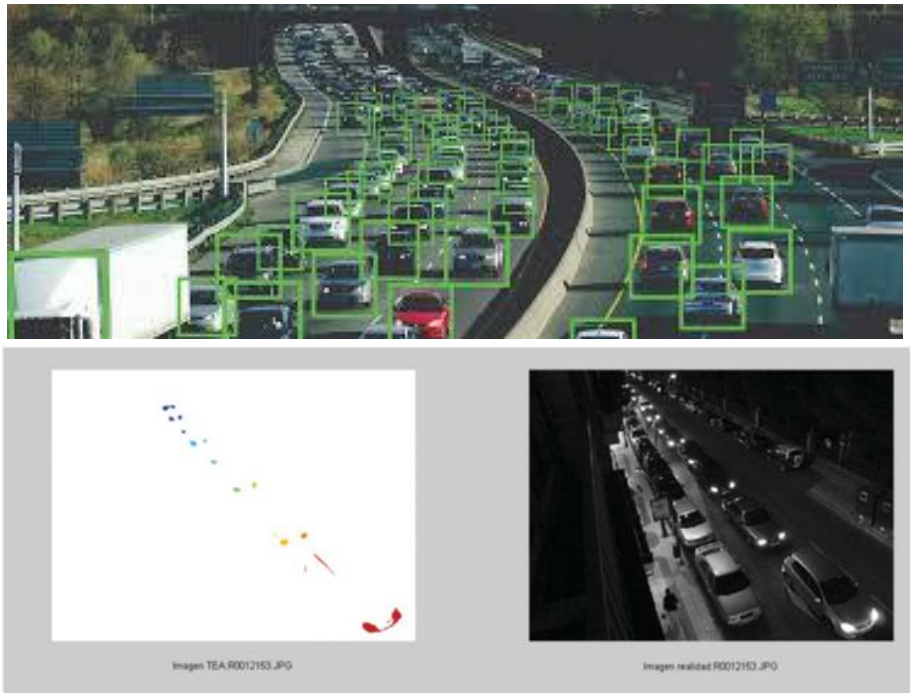


2.2. ƯỚC LƯỢNG XE CỘ

Bài toán ước lượng xe cộ thường được bao bọc một bài toán lớn hơn là quản lý giao thông thông minh. Quản lý giao thông thông minh bao gồm ước lượng xe cộ, phát hiện vi phạm giao thông, phân luồng tự động ,...

a. NGOÀI NƯỚC

- Hệ thống ước lượng xe cộ sơ khởi đã xuất hiện từ trước 2010. Nhưng lúc đó các hệ thống xử lý rất cồng kềnh và vất vả.
- Nhưng tốc độ phát triển thì lại rất rất nhanh. Hiện tại các thành phố lớn trên thế giới đều đã được trang bị hệ thống quản lý giao thông thông minh.
- Độ chính xác hiện tại của các hệ thống đã gần như tuyệt đối, các nghiên cứu mới chỉ góp phần làm giảm thời gian xử lý và tăng thêm khả năng xử lý trong các điều kiện khác nhau.
- Đa phần đều làm oto và không có moto. Chỉ một số quốc gia có moto mới tập trung phát triển hệ thống cho moto (Trung Quốc, Đài Loan, Singapore, Việt Nam, Thái Lan,...)



→ Tóm lại hiện nay hệ thống ước lượng mật độ đám đông , hệ thống ước lượng xe cộ trên thế giới đã "làm ra đầu ra đũa". Còn tại Việt Nam đa phần các hệ thống mua lại api từ nước ngoài, chưa có hệ thống áp dụng thực tế nào "Design in VietNam" 100%.

b. TRONG NƯỚC

- Năm 2015, nhóm nghiên cứu của Công ty TNHH Thế hệ Geo và Công ty Trí tuệ Nhân tạo VedaX đã bắt đầu nghiên cứu và xây dựng thí điểm ứng dụng công nghệ trí thông minh nhân tạo trong quản lý và điều khiển giao thông tại Việt Nam

- Đây là hệ thống đã có từ năm 2015 (nghĩa là đã được 5 năm). Hệ thống trên hình được áp dụng tại khu vực ngã tư hàng xanh. Điều đặc biệt là hệ thống đã nhận diện được xe máy. Đây là một điều rất đặc biệt vì xe máy chỉ phổ biến ở một số quốc gia nhất định trong đó có VN.Hệ thống hoạt động được cả trong thời tiết xấu
- Link video demo hệ thống TRANSPOX: https://youtu.be/PuvaZxmZeuM
- Ngoài ra hiện tại đa phần các thành phố lớn tại VN đã áp dụng hệ thống giao thông thông minh như Hà Nội, TP.HCM, Đà Nẵng, Huế, Cần Thơ,... Tất cả hệ thống hiện nay đều hoạt động dược trong nhiều dạng thời tiết và góp phần không nhỏ cho việc quy hoạch giao thông, phân luồng giao thông, kiểm tra phát hiện vi phạm giao thông.

3.PHÁT BIỂU BÀI TOÁN

3.1. NGUYÊN LÝ

- Tại mỗi điểm ảnh sẽ tính xác suất xuất hiện đầu người. Sau đó sẽ tính tổng xích ma xác suất của tất cả điểm ảnh có trong ảnh.

3.2. INPUT/OUTPUT

- **Input :** Ảnh tĩnh RGB
- **Output:** Ảnh chứa xác suất xuất hiện đối tượng trong ảnh (đầu người, xe, …) (density map).
  Từ đó tìm được ma trận chứa số lượng đối tượng trong từng phân vùng ảnh (count map).
  Dùng count map để ước lượng được số lượng chính xác hơn so với chỉ dùng density map.

3.3. GIỚI HẠN BÀI TOÁN

- Ước lượng mật độ đám đông (con người hoặc phương tiện giao thông) trong các hình ảnh tĩnh.

3.4. THÁCH THỨC BÀI TOÁN

- Trong mỗi ảnh có thể có mật độ thưa, vừa, dày đặc, hệ thống phải thích ứng như thế nào (xử lý tốt) với cùng cả ba mật độ như vậy.
- Hình bị biến dạng do phép chiếu phối cảnh: những đầu người gần cam thì to và càng nhỏ dần khi xa camera.
- Sự biến thiên về mật độ trong cùng một ảnh (ví dụ: góc trái: thưa, góc giữa: vừa, góc phải: đông)

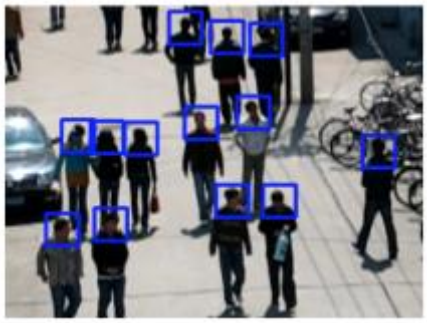
  → Hệ thống nào đáp ứng thách thức càng nhiều thì càng tốt.

4.CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

4.1. ĐẾM DỰA TRÊN ƯỚC LƯỢNG ĐỐI TƯỢNG

- **Nội dung**: Với vấn đề đếm thì hướng giải quyết dễ thấy đầu tiên và đơn giản nhất đó là phát hiện ra đối tượng và đếm nó.
- **Một số đại diện**:
  - Phát hiện toàn bộ (Monolithic detection) : Đếm bằng cách phát hiện toàn bộ cơ thể người trong ảnh. Hướng tiếp cận là dùng 1 cửa sổ trượt (sliding window) để phát hiện đối tượng và đếm.

  - Phát hiện bộ phận (Part-based detection): Thay vì phát hiện toàn bộ cơ thể người thì phương pháp này tập trung phát hiện bộ phận đặc trưng nhất, ví dụ đối với con người thì đầu người, xe là bánh xe), từ đó ước lượng số người/xe trong hình. Tuy nhiên ta thường kết hợp nhiều bộ phận, ví dụ như kết hợp đầu và vai thì hiệu quả hơn so với chỉ phát hiện đầu người



- **Thách thức**:
  - Phát hiện toàn bộ: Phương pháp này chỉ áp dụng đối với trường hợp các đối tượng rời rạc và rõ ràng trong ảnh input. Đối với trường hợp các đối tượng xuất hiện một cách lộn xộn và bị che khuất thì phương pháp này thiếu hiệu quả và độ chính xác thấp.
  - Phát hiện bộ phận: Giải quyết vấn đề các đối tượng bị che khuất. So với phương pháp phát hiện toàn bộ đối tượng thì phương pháp này cải thiện độ chính xác trong môi trường phức tạp.

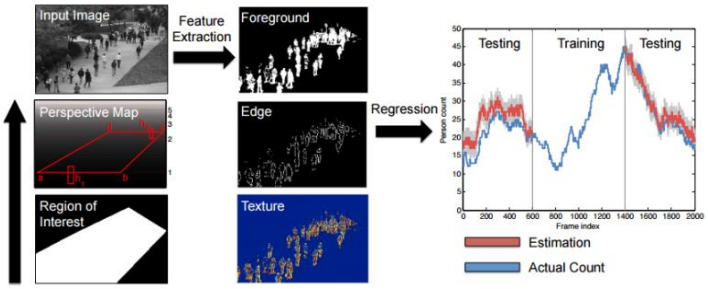
→ Tuy nhiên, nhìn chung phương pháp ước lượng dựa trên phát hiện đối tượng riêng lẻ vẫn không hiệu quả trong trường hợp ảnh có mật độ đông và background (cây cối, chim, bảng hiệu, …) phức tạp.

4.2. ĐẾM DỰA TRÊN MÔ HÌNH HỒI QUY

- Giải quyết vấn đề gì: Phương pháp này đưa ra nhằm giải quyết các thách thức trong các phương pháp phát hiện ra đối tượng riêng lẻ. Ước lượng mật độ đám đông dựa trên mô

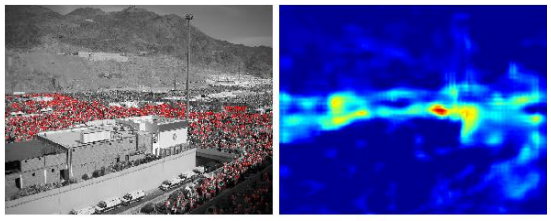
tả/đặc trưng tổng quát được rút trích từ toàn bộ mô hình đám đông. Do không đòi hỏi phải phân biệt rõ ràng hay theo dõi các đối tượng một cách riêng lẻ nên phương pháp này được áp dụng khá tốt trong đám đông với mật độ lớn và có phân bố phức tạp.

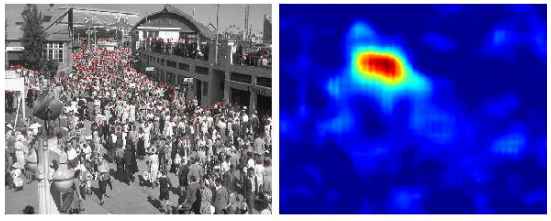
- Nội dung: Sử dụng các đặc trưng cấp thấp để ước lượng mật độ đám đông. Đầu tiên, ta xác định khu vực quan tâm cần tính toán và tìm ra sơ đồ chuẩn hóa (normalization map). Sau đó sẽ rút trích những đặc trưng tổng thể và huấn luyện ra mô hình hồi quy sử dụng những đặc trưng đã được chuẩn hóa.
- Online: Rút trích các đặc trưng của ảnh input và áp dụng mô hình hồi quy để ước lượng số lượng thực tế.
- Offline: Đầu tiên, người ta sẽ tiến hành trích xuất những đặc trưng cấp thấp (low-level features) như là các pixel ở tiền cảnh (foreground) và các đặc trưng cạnh (edge details). Sau đó kết hợp các hợp các đặc trưng này với số lượng thực tế để huấn luyện mô hình hồi quy.



4.3. ƯỚC LƯỢNG DỰA TRÊN DENSITY MAP

- Giải quyết vấn đề gì: Phương pháp này đưa ra nhằm giải quyết các thách thức trong các phương pháp phát hiện ra đối tượng không phải riêng lẻ nữa mà là một cụm người.
- Nội dung: Thay vì ước lượng trực tiếp số lượng đối tượng thì ta tiếp cận một cách gián tiếp bằng cách dùng mô hình hồi quy để trả về density map cho ảnh input, sau đó từ density map để tính được số lượng đối tượng.
- Offline:
    - Chuẩn bị ground truth: Từ ảnh ban đầu, người ta chấm các điểm đánh dấu lên đối tượng (ví dụ bên dưới là chấm các điểm màu đỏ lên đầu người). Sau đó dùng hàm Gaussian để từ các điểm đánh dấu tạo thành density map.


Actual Count: 1115 Estimated: 1143

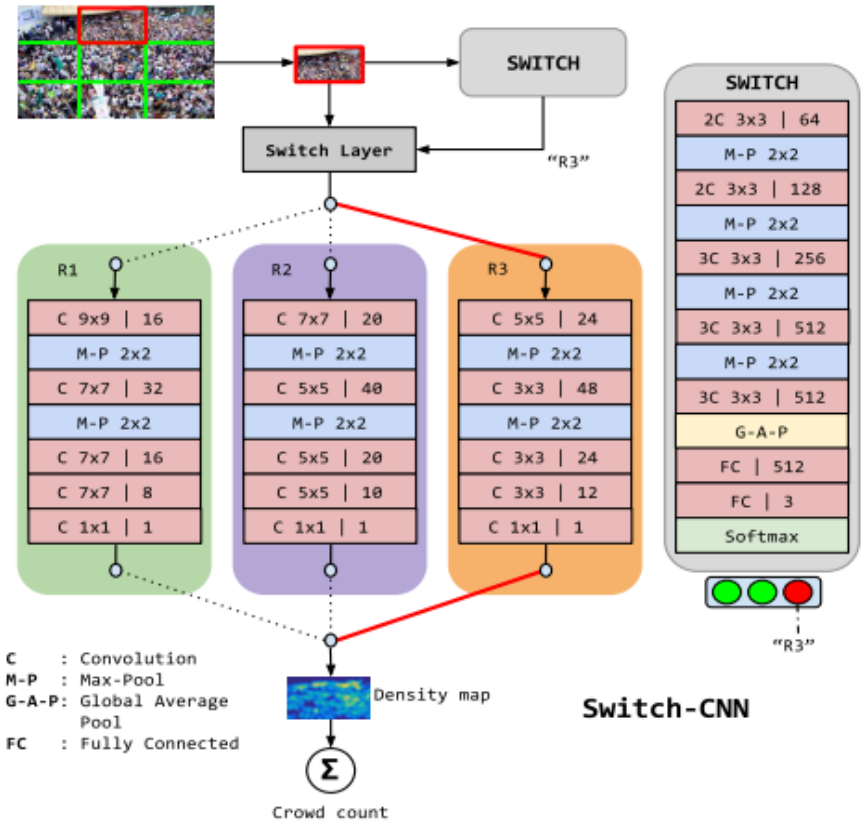

Actual Count:440 Estimated:433

- Từ các cặp dữ liệu (hình ảnh, density map) ta huấn luyện mô hình hồi quy.

- Online:
  - Áp dụng mô hình hồi quy để từ ảnh đầu về trả về density map.
  - Sau đó ước lượng số đối tượng bằng cách lấy tổng giá trị các điểm trong density map.
- Thách thức: Trong một ảnh mật độ không đồng đều nhau, có chỗ đông chỗ thưa; vị trí đối tượng ở gần ở xa khung hình,... thì phương pháp này hoạt động chưa thực sự hiệu quả.

4.4. SWITCH CNN

- Giải quyết vấn đề gì: Trong một ảnh mật độ không đồng đều nhau, có chỗ đông chỗ thưa; vị trí đối tượng ở gần ở xa khung hình,...
- Nội dung: Switch CNN có 3 mô hình hồi quy R1, R2, R3. Switch CNN hoạt động bằng cách chia ảnh input thành nhiều patch, mỗi patch được đưa qua 1 bộ phân lớp để xác định R1, R2 hay R3 là thích hợp nhất.
- Input/Output:
  - Input mạng 1: gồm các patches R1, R2, R3
  - Output mạng 1: đưa ảnh vào và gán các patches tương ứng với ảnh
  - Input mạng 2: các patches, các density map đã được đánh dấu đầu người
  - Output mạng 2: density map
  - Online: khi switch trả về lable nào (R1, R2, R3) thì sẽ rẽ nhánh sang các mạng R1, R2, R3 tương ứng.



4.5. ƯỚC LƯỢNG DỰA TRÊN "CHIA ĐỂ TRỊ"

- Giải quyết vấn đề gì: Với ảnh mật độ cao và có chỗ đông chỗ thưa; vị trí đối tượng ở gần ở xa khung hình, ...
- Nội dung: Thay vì xem số lượng đối tượng là một giá trị liên tục thì S-DCNet rời rạc hoá bằng cách chia số lượng thành các khoảng (gọi là count level) và dùng phương pháp classification để xác định ảnh thuộc khoảng nào.
- Input/Output:
    - Input: Ảnh đầu vào I có kích thước MxN và số lần chia K (số tầng trong SDC-Net)
    - Output: Số lượng đối tượng trong ảnh C

4.6. BẢNG SO SÁNH THỰC NGHIỆM GIỮA CÁC PHƯƠNG PHÁP

- Với bộ dữ liệu UCF CC 50 : là bộ dữ liệu chứa các hình ảnh đám đông với mật độ cực kì
- dày đặc. Các hình ảnh này được thu thập từ FLICKR.

Bài báo	*Phương pháp*	*MAE*	*Year*
From Open Set to Closed Set: Counting Objects by Spatial Divide-and-Conquer	S-DCNET	204.2	2019
Context-Aware Crowd Counting	CAN	212.2	2018
Locate, Size and Count: Accurately Resolving People in Dense Crowds via Detection	LSC-CNN	225.6	2019
Scale Aggregation Network for Accurate and Efficient Crowd Counting	SANet	258.4	2018
Switching Convolutional Neural Network for Crowd Counting	Switch-CNN	318.1	2017

- Với bộ dữ liệu TRANCOS : bộ dữ liệu được thu thập từ các video giám sát giao thông ở thành phố Dirección General de Tráfico Tây Ba Nha.

Method	GAME(0)	GAME(1)	GAME(2)	GAME(3)
CCNN [25]	12.49	16.58	20.02	22.41
Hydra-3s [25]	10.99	13.75	16.69	19.32
CSRNet [20]	3.56	5.49	8.57	15.04
SPN [8]	3.35	4.94	6.47	9.22
S-DCNet	**2.92**	**4.29**	**5.54**	**7.05**

5. PHƯƠNG PHÁP

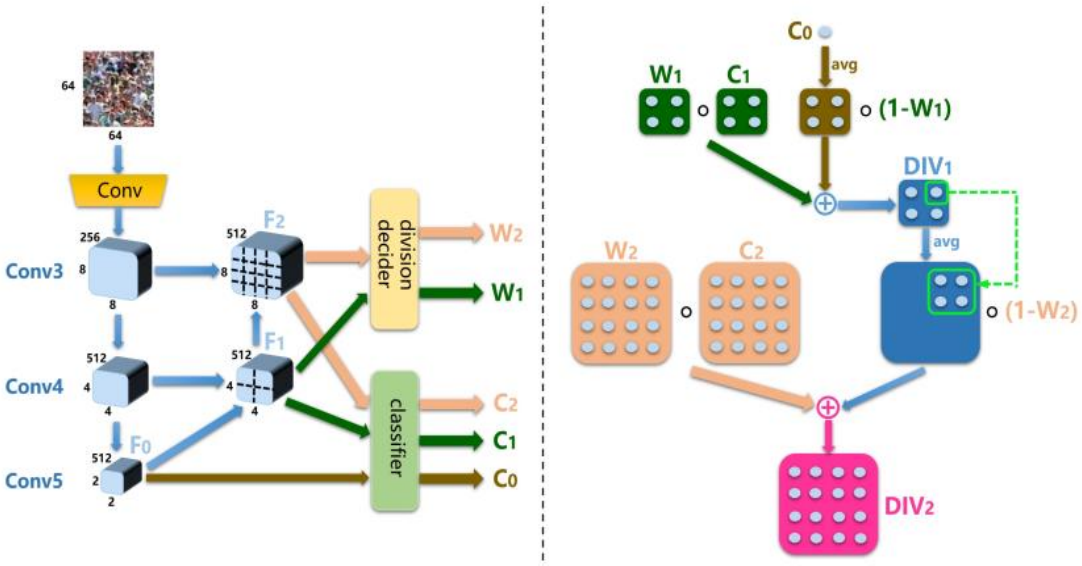
5.1. RỜI RẠC HÓA SỐ LƯỢNG ĐỐI TƯỢNG

- Các phương pháp trước thường dùng hồi quy để trả về số lượng đối tượng trong ảnh hoặc trả về density map. Từ density map này ta tính tổng giá trị pixel để được số lượng đối tượng trong ảnh.
- S-DCNet tiếp cận theo một hướng khác, thay vì xem số lượng đối tượng là một giá trị liên tục thì S-DCNet rời rạc hoá bằng cách chia số lượng thành các khoảng (gọi là count level) và dùng phương pháp classification để xác định ảnh thuộc khoảng nào.

- Ví dụ, ở ảnh trên density map là cách tiếp cận truyền thống. Từ density map ta lấy tổng giá trị từng patch và tính được count map. Từ count map ta quy về khoảng để được class map.
- Cách chia khoảng được mô tả như sau:
  - Ta thấy, số lượng đối tượng trong một ảnh sẽ không có chặn trên và thuộc khoảng $[0, +\infty)$
  - Chia tập $[0, +\infty)$ thành các khoảng sau: $\{0\}$, $(0, C_1]$, $(C_2, C_3]$, ... , $(C_{M-1}, C_M]$ và $(C_M, +\infty)$ được gán nhãn từ 0 đến M. Trong đó CM không lớn hơn số lượng cục bộ tối đa trong tập training.
  - Ví dụ như số lượng rơi vào khoảng $(C_2, C_3]$ thì nó sẽ được gán nhãn là 2
  - Ngược lại, để quy đổi từ khoảng thành số lượng, ta lấy trung vị của từng khoảng. Đối với khoảng $(C_M, +\infty)$ thì C_M sẽ được lấy làm số lượng.
  - Ví dụ khoảng (1, 2] thì được quy đổi là 1.5; khoảng $(20, +\infty)$ được quy đổi là 20.

5.2. CẤU TRÚC CỦA SDC-NET

- SDC-Net bao gồm VGG16 feature encoder, UNet-like decoder, Classifier và Division decider.
- Trong đó mỗi thành phần có các chức năng riêng biệt:
  - VGG16 feature encoder: từ một patch tính toán ra feature map.
  - UNet-like decoder: upsample kích thước feature map.
  - Classifier: phân lớp patch thuộc khoảng bao nhiêu đối tượng.



(Minh hoạ cấu trúc SDC-Net 2 tầng với ảnh đầu vào kích thước 64X64)

a. VGG16 feature encoder

- Có chức năng rút trích feature map cho ảnh đầu vào.
- Input: Ảnh I.
- Output: Feature map F.

b. UNet-like decoder

- Có chức năng upsample feature map đầu vào và trả về feature map mới có kích thước gấp đôi.
- Input: Feature map F có kích thước M x N.
- Output: Feature map F' có kích thước 2M x 2N

c. Classifier

- Có chức năng phân lớp, gán nhãn cho patch thuộc khoảng bao nhiêu đối tượng.
- Input: Feature map F, gồm các vùng i, i=1, 2, ...N.
- Output: Ma trận nhãn CLS có N phần tử, trong đó mỗi phần tử chứa nhãn về khoảng số lượng trong vùng F(i).
- Kiến trúc Classifier được mô tả trong bảng sau, theo format:
  - Conv size x size, output channel, s stride.
  - Mỗi lớp convolution được theo sau bởi hàm ReLU ngoại trừ lớp cuối.

classifier
2×2 AvgPool, s 2
1×1 Conv, 512, s 1
1×1 Conv, $class\ num$, s 1
–

d. Division decider

- Có chức năng quyết định xem vị trí nào cần chia ra để tăng độ chính xác khi tính số lượng.
- Input: Feature map F, gồm các vùng i, i=1, 2, ...N.
- Output: Ma trận division mask W có N phần tử, trong đó mỗi phần tử là xác suất cần chia vùng F(i) ra.
- Kiến trúc Classifier được mô tả trong bảng sau, theo format:
  - Conv size x size, output channel, s stride.
  - Mỗi lớp convolution được theo sau bởi hàm ReLU ngoại trừ lớp cuối.

division decider
2×2 AvgPool, s 2
1×1 Conv, 512, s 1
1×1 Conv, 1, s 1
Sigmoid

5.3. THUẬT TOÁN SDC-NET ĐA TẦNG

- **Input**: Ảnh đầu vào I có kích thước MxN
  Số lần chia K (hay nói cách khác là số tầng trong mô hình S-DC)
- **Output**: Số lượng đối tượng trong ảnh C
- Thuật toán các bước:

- **Bước 1**: Từ ảnh I qua VGG16 rút trích được feature map F_0 với kích thước (M/32)x(N/32)x512.
- **Bước 2**:
  - Dùng Classifier để phân loại nhãn của F_0, ký hiệu là CLS_0 (CLS_0 cho ta biết số lượng đối tượng trong F_0 thuộc khoảng giá trị nào).
  - Sau đó từ CLS_0 ta tính giá trị C_0 (số lượng đối tượng) theo quy ước ở mục 1 (Rời rạc hoá số lượng đối tượng)
- **Bước 3**: Khởi tạo $DIV_0=C_0$
- **Bước 4**: Lặp K lần: Gán i=1
- **Bước 5**:
  - Đưa feature map F_{i-1} vào UNet-like decoder để thu được F_1. UNet-like decoder sẽ upsample kích thước F_{i-1} lên gấp 4 lần.
  - Ví dụ: F_0 có kích thước 2x2x512 được xem như 1 vùng. Qua UNet-like decoder thì F_1 có kích thước 4x4x512 và xem như 4 vùng. Tương tự F_2 có kích thước 8x8x512 và được phân thành 16 vùng
- **Bước 6**: Từ feature map F_i:
  - Dùng Classifier để tìm được ma trận CLS_i
    - F_i được chia thành các phần có kích thước bằng với F_0, sau đó đưa vào Classifier để thu được ma trận CLS_i
    - CLS_i có kích thước bằng với số vùng của F_i, trong đó mỗi phần tử xác định nhãn của vùng tương ứng.
    - Ví dụ: Với ảnh input 64x64 thì F_1 là feature map ứng với 4 vùng của ảnh (mỗi vùng 32x320). CLS_1 có kích thước 2x2, mỗi phần tử chứa giá trị nhãn của vùng đó.
  - Dùng division decider để tìm được ma trận devision mask W_i
    - W_i có kích thước bằng với CLS_i, trong đó mỗi phần tử w có giá trị thuộc [0, 1] và cho biết tại vị trí đó có cần chia ra để tính hay không.
    - w = 0 nghĩa là không cần thực hiện phép chia tại vị trí này.
    - w = 1 nghĩa là cần chia ra để tính tại vị trí này nhằm tăng độ chính xác.
- **Bước 7**: Từ CLS_i ta tính C_i theo quy ước ở mục 1 (Rời rạc hoá số lượng đối tượng).
- **Bước 8**:
  - Cập nhật DIV ở bước thứ i theo công thức sau:

$$DIV_i = (\mathbb{1} - W_i) \circ avg(DIV_{i-1}) + W_i \circ C_i \,.$$

  - Bước này có ý nghĩa là dựa vào ma trận xác suất W_i để xác định vị trí nào trong ảnh ưu tiên dùng lại giá trị cũ (DIV_{i-1}) và vị trí nào ưu tiên dùng giá trị sau khi chia (C_i)
  - Trong đó:
    - "$\circ$" ký hiệu phép nhân Hadamard

- **avg** là toán tử tái phân phối theo trung bình, bằng cách upsample ma trận lên 4 lần, với mỗi phần tử trong ma trận cũ trở thành 1 vùng 2x2 trong ma trận mới.

Ví dụ: DIV_{i-1}

x	y
z	t

$avg(DIV_{i-1})$

x/4	x/4	y/4	y/4
x/4	x/4	y/4	y/4
z/4	z/4	t/4	t/4
z/4	z/4	t/4	t/4

  - Quay lại bước 4 nếu i<K.
  - Ngược lại đến bước 9.
- Bước 9: Tính tổng các giá trị trong ma trận DIV_N để thu được số lượng đối tượng C. Trả về C và kết thúc thuật toán.

Algorithm 1: Multi-Stage S-DC

Input: Image I and division time N
Output: Image count C

Extract F_0 from I;
Generate CLS_0 given F_0 with the classifier, and recover C_0 from CLS_0;
Initialize $DIV_0 = C_0$;
for $i \leftarrow 1$ ***to*** N **do**
    Decode F_{i-1} to F_i;
    Process F_i with the classifier and the division decider to obtain CLS_i and the division mask W_i;
    Recover C_i from CLS_i;
    Update DIV_i as per Eq. 2 ;
Integrate over DIV_N to obtain the image count C;
return C

5.4. GIAI ĐOẠN OFFLINE

- Chuẩn bị ground truth:
  - Density map: Mỗi ảnh trong dataset đã được đánh dấu bằng cách chấm lên mỗi đối tượng. Sau đó dùng hàm Gaussian để chuyển ảnh thành density map
  - Count map: Count map được tính bằng cách lấy tổng giá trị pixel trong mỗi patch của density map, dựa trên công thức:

$$N_p(i) = \sum_x D(x) \quad x \in p(i)$$

    - Trong đó:

- Np là count map.
- p(i) là tập hợp pixel trong patch thứ i.
- D là density map.

- Huấn luyện mô hình theo giải thuật ở đã được trình bày.

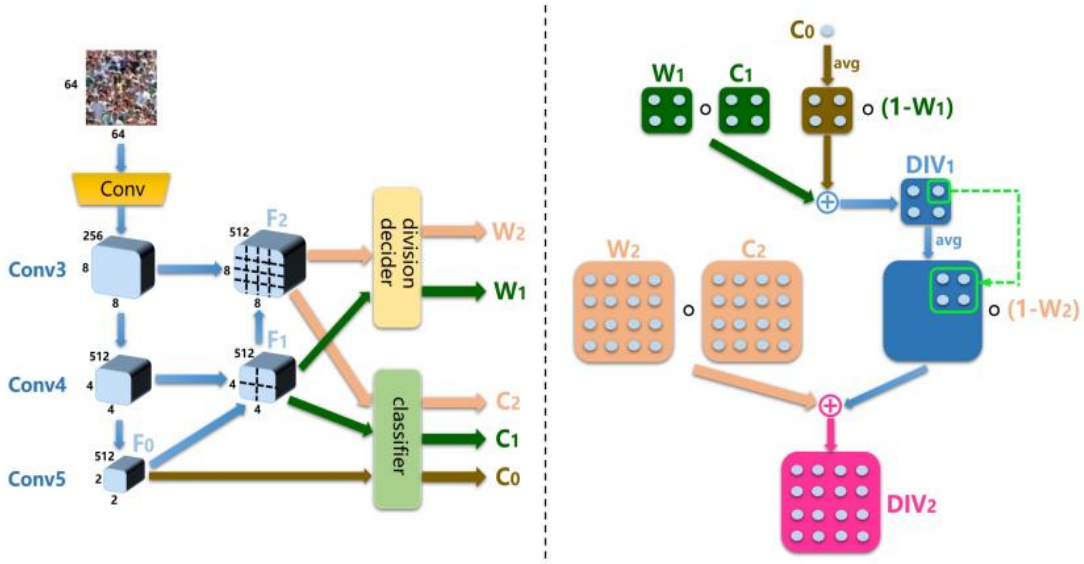
5.5. GIAI ĐOẠN ONLINE

- Áp dụng mô hình được cài đặt để từ ảnh đầu vào trả về density map
- Chuyển density map thành count map
- Từ count map sẽ ước lượng số lượng đám đông

6. CÀI ĐẶT CHƯƠNG TRÌNH

6.1. Tiến hành cài đặt mạng SDC-Net theo mô hình

- Tên file: SDCNet.py



- Bước 1: Tạo các module cơ sở
  - Hàm tạo các layer VGG để tham gia vào mô hình mạng (conv1, conv2, conv3, conv4, conv5)
    - *Tên hàm : make_layers(cfg, in_channels = 3,batch_norm=False,dilation = False)*
    - *Ví dụ : conv1_features = make_layers([64, 64, 'M'], in_channels=3)*
      => dùng để lưu conv1 features với kích thước 64x64
- Bước 2: Xây dựng các hàm để áp dụng vào quá trình **upscampling** (tăng kích thước các layer trong mạng)
  - Tên hàm: one_conv, double_conv, three_conv
- Bước 3: Xây dựng class UNet-stype upscampling
  - Tên class: up()
- Bước 4: Xây dựng kiến trúc mạng SDCNet VGG16:
  - **Tên class**: SDCNet_VGG16_classify, các bước cơ bản
  - Khởi tạo các thành phần cơ bản của mạng như kích thước các conv_feautures, activation function (dùng ReLu), load wrights của mô hình VGG16. **tên hàm: __init__**

- Quá trình truyền thẳng (forward) để tìm được các feature_map. **tên hàm**: **forward**
- Khởi tạo trọng số w, **tên hàm**: **_initialize_weights**
- Lan truyền ngược để tìm được các trọng số (W2, W1, C0,C1,C2), **tên hàm**: **resample**
- Cập nhật DIV theo công thức:

$$DIV_i = (\mathbb{1} - W_i) \circ avg(DIV_{i-1}) + W_i \circ C_i \,.$$

Tên hàm: parse_merge

6.2. Các hàm cài đặt

Các function chính

- Func1: Chuyển density map thành count map
  density map = batch size * 1 * w * h
  Tên hàm: get_local_count(density_map,psize,pstride)
- Func2: convert count to class (0->c-1)
  Tên hàm: Count2Class(count_map,label_indice)
- Func3: convert class (0->c-1) to count number
  Tên hàm: Class2Count(pre_cls,label_indice)
  - Input:
    - pre_cls là class label range in [0,1,2,...,C-1]
    - label_indice not include 0 but the other points
  - Output: count value, the same size as pre_cls

6.3. Cấu trúc chương trình

- Cấu trúc chương trình bao gồm:
  (Nguồn tham khảo: https://github.com/xhp-hust-2018-2011/S-DCNet)
  - Folder 'model': chứa model đã được train sẵn
  - Folder 'Network': chứa các file cấu trúc mạng SDCNet (class_func.py, merge_func.py, SDCNet.py)
  - Folder 'Test_Data': chứa các tập dữ liệu để test
    - Sample_One: folder dùng để test ảnh bất kỳ
    - SH_partA_Density_map, SH_partB_Density_map: folder chứa tập dữ liệu ShanghaiTech
  - Iotools.py: write file .txt (log.txt)
  - load_data_V2.py: chuyển dữ liệu từ hình ảnh sang dataset của pytorch
  - main_process.py: phần xử lý chính
  - SHAB_main.py: dùng để test dữ liệu của bộ dataset ShanghaiTech
  - One Sample.py (tự thêm vào): dùng để test ảnh bất kỳ
  - LICENSE
  - Requirements.txt
  - README.TXT: hướng dẫn sử dụng + hướng dẫn cài đặt
- Hướng dẫn cài đặt
  - Ngôn ngữ lập trình Python 3.6
  - IDE khuyến nghị: PyCharm Community Edition
    Link tham khảo: https://www.jetbrains.com/pycharm/download/

- Requirements
  + python==3.6.2
  + pytorch>=0.4.0
  + numpy==1.14.0
  + scikit-image==0.13.1
  + scipy==1.0.0
  + pandas==0.22.0

- Hướng dẫn sử dụng
  - Để chạy test trên bộ dữ liệu ShanghaiTech: Run file SHAB_main.py
  - Để chạy test các ảnh khác (chỉ sử dụng ảnh jpeg (*.jpg))
    - Chuyển ảnh vào thư mục: Test_Data/Sample_One/test/images
    - Thay đổi tên thành dạng IMG_x.jpg
    - Chạy file One Sample.py

7. THỰC NGHIỆM

7.1. Thực nghiệm với tập SHA, SHB:

a. Tập test SHA:

```
Test:[  172/  182] pre: 281.203 gt:371.000 err:-89.797 frame: 0.71Hz/1.04Hz
Test:[  173/  182] pre: 1733.691 gt:2256.000 err:-522.309 frame: 0.76Hz/1.04Hz
Test:[  174/  182] pre: 92.518 gt:101.000 err:-8.482 frame: 0.60Hz/1.04Hz
Test:[  175/  182] pre: 1343.716 gt:1366.000 err:-22.284 frame: 0.80Hz/1.04Hz
Test:[  176/  182] pre: 232.672 gt:255.000 err:-22.328 frame: 2.05Hz/1.04Hz
Test:[  177/  182] pre: 60.738 gt:69.000 err:-8.262 frame: 2.07Hz/1.05Hz
Test:[  178/  182] pre: 149.413 gt:190.000 err:-40.587 frame: 2.88Hz/1.06Hz
Test:[  179/  182] pre: 216.867 gt:246.000 err:-29.133 frame: 1.95Hz/1.06Hz
Test:[  180/  182] pre: 486.643 gt:521.000 err:-34.357 frame: 0.59Hz/1.06Hz
Test:[  181/  182] pre: 141.821 gt:153.000 err:-11.179 frame: 1.35Hz/1.06Hz
Test:[  182/  182] pre: 161.818 gt:242.000 err:-80.182 frame: 1.22Hz/1.06Hz
                mae    &    rmse   &      me  \\
test          57.575      98.093     -18.797
```

- Ta được các giá trị:
  - sai số toàn phương trung bình **MAE** : 57.575 (người)
  - Sai số trung bình bình phương **RMSE :** 98.093
  - Trung bình sai số **ME :** -18.797

b. Tập test SHB:

```
Test:[  308/  316] pre: 88.187 gt:92.000 err:-3.813 frame: 0.35Hz/0.48Hz
Test:[  309/  316] pre: 64.827 gt:64.000 err:0.827 frame: 0.36Hz/0.48Hz
Test:[  310/  316] pre: 209.003 gt:217.000 err:-7.997 frame: 0.35Hz/0.48Hz
Test:[  311/  316] pre: 48.613 gt:53.000 err:-4.387 frame: 0.36Hz/0.48Hz
Test:[  312/  316] pre: 50.255 gt:48.000 err:2.255 frame: 0.35Hz/0.48Hz
Test:[  313/  316] pre: 238.825 gt:211.000 err:27.825 frame: 0.35Hz/0.48Hz
Test:[  314/  316] pre: 123.087 gt:107.000 err:16.087 frame: 0.35Hz/0.48Hz
Test:[  315/  316] pre: 249.851 gt:249.000 err:0.851 frame: 0.63Hz/0.48Hz
Test:[  316/  316] pre: 120.559 gt:113.000 err:7.559 frame: 0.58Hz/0.48Hz
                mae    &    rmse   &      me  \\
test           6.633      10.813       1.294
```

- Ta được các giá trị:
  - Sai số toàn phương trung bình **MAE** : 6.633 (người)
  - Sai số trung bình bình phương **RMSE :** 10.813
  - Trung bình sai số **ME :** 1.294

7.2. Thực nghiệm với ảnh chụp tập thể thực tế:

a. Ảnh có kết cấu đơn giản (không có nhiều nhiễu)

- **Ảnh input:**

- **Chạy Chương trình ước lượng, kết quả dự đoán: 43.832 [d]**
- **Đếm thủ công bằng tay : 43**

- Sai số tương đối = (43.832 - 43)/43 = 0.01934 = 1.934%

Nhận xét: Với bố cục ảnh như trên, thay vì ngồi đếm bằng cách chấm từng điểm thì ta có thể đưa ảnh vào mô hình để đưa ra giá trị ước lượng khá chính xác

b. Ảnh có mật độ đám đông dày đặt ở một khu vực

Input:

Ước lượng bởi chương trình: 34.794 [d]

Giá trị khi đếm thủ công: 39

Sai số tương đối: |34.794-39|/39 = 0.1078 = 10.78%

Nhận xét: Có thể thấy mô hình hoạt động không thật sự tốt cho trường hợp này.

c. Ảnh bị đám đông bị che khuất do góc chụp

Input:

Ước lượng bởi chương trình: 31.417 [d]

Giá trị khi đếm thủ công: 32

Sai số tương đối = |31.417-32|/32 = 0.0164 = 2.64%

Nhận xét: Ta có thể thấy mô hình thực hiện tương đối tốt với bố cục hình ảnh như trên, ta có thể áp dụng mô hình thay vì đếm tay.

d. Kết quả khi chạy chương trình với 3 ảnh input trên:

```
Begin to test for Sample_One
Test:[    1/    4] pre: 43.832 gt:172.000 err:-128.168
Test:[    2/    4] pre: 34.794 gt:1111.000 err:-1076.206
Test:[    3/    4] pre: 31.417 gt:301.000 err:-269.583
```

8. TÀI LIỆU THAM KHẢO

- Haipeng Xiong, Hao Lu, Chengxin Liu, Liang Liu, Zhiguo Cao, Chunhua Shen, "**From Open Set to Closed Set: Counting Objects by Spatial Divide-and-Conquer**", Computer Vision and Pattern Recognition, 2019
- Guangshuai Gao, Junyu Gao, Qingjie Liu, Qi Wang3, and Yunhong Wang, "**CNN-based Density Estimation and Crowd Counting: A Survey**", 2020
- Yuhong Li, Xiaofan Zhang, and Deming Chen, "**Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes**", In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
- Deepak Babu Sam, Shiv Surya, R. Venkatesh Babu, "**Switching Convolutional Neural Network for Crowd Counting**", Computer Vision and Pattern Recognition, 2017
- Hao Lu, Zhiguo Cao, Yang Xiao, Bohan Zhuang, and Chunhua Shen, "**TasselNet: counting maize tassels in the wild via local counts regression network**", Plant Methods, 2017.
- Joseph Paul Cohen, Genevieve Boucher, Craig A. Glastonbury, Henry Z. Lo, and Yoshua Bengio, "**Count-ception: Counting by fully convolutional redundant**

counting", In Proc. IEEE International Conference on Computer Vision Workshop (ICCVW), 2017.

- Tobias Stahl, Silvia L Pintea, and Jan C van Gemert, "**Divide and count: Generic object counting by image divisions**", IEEE Transactions on Image Processing, 2019.
- Karen Simonyan and Andrew Zisserman, "**Very deep convolutional networks for large-scale image recognition**", Computer Science, 2014.