Deep learning of Vietnamese traditional foods for real-time recognition: VNTF22

Nguyễn Hoàng Minh[1,2,3], Nguyễn Thiện Thuật[1,2,3], Tạ Nhật Minh[1,2,3], Nguyễn Minh Tiến[1,2,3], and Đỗ Trọng Hợp[1,2,4]

[1] University of Information Technology, Ho Chi Minh, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
[3] {20521609,20521998,20521614,20522010}@gm.uit.edu.vn
[4] hopdt@gm.uit.edu.vn

Tóm tắt nội dung Trong bài viết này, chúng tôi đề xuất sử dụng các thuật toán học sâu để nhận dạng các món ăn truyền thống Việt Nam. Chúng tôi sử dụng các mô hình one-stage như You Only Look Once (YOLO) v7, SSD và RetinaNet để đáp ứng yêu cầu cao về thời gian thực và độ chính xác. Chúng tôi cũng đóng góp một bộ dữ liệu đã được xử lý và gán nhãn để đánh giá hiệu suất của các mô hình. Kết quả cho thấy rằng, sử dụng mô hình one-stage để nhận dạng các món ăn truyền thống tại thời điểm thực có thể đưa ra nhiều kết quả đúng trong cùng một khung hình.

Keywords: Deep-learning · Vietnamese food recognition· YOLOv7· SSD· RetinaNet· one-stage model· Vietnamese food dataset.

1 Giới thiệu

Trong xu thế công nghệ thực phẩm hiện đại, nhận dạng thức ăn là một bài toán quan trọng với rất nhiều ứng dụng trong việc quản lý và đánh giá chất lượng các món ăn. Du khách từ khắp nơi trên thế giới đến Việt Nam để trải nghiệm những điểm du lịch tuyệt vời. Tuy nhiên, việc tìm hiểu về các loại bánh truyền thống của từng địa phương có thể khó khăn vì mỗi nơi có những loại bánh riêng biệt và tên gọi không giống nhau. Thêm vào đó, các loại bánh còn có thể thay đổi màu sắc và hình dáng theo từng khu vực. Vấn đề còn lớn hơn là không biết nên chọn loại bánh nào để làm quà lưu niệm hoặc thưởng thức. Để giải quyết bài toán này, việc sử dụng kỹ thuật học sâu trở nên phổ biến và được áp dụng rộng rãi. Trong bài báo nghiên cứu của chúng tôi, chúng tôi sẽ tập trung vào việc nhận dạng thức ăn truyền thống của Việt Nam bằng cách sử dụng kỹ thuật học sâu. Nhận dạng thức ăn là một trong những nền tảng của việc xử lý hình ảnh và video, sử dụng kỹ thuật học sâu cho việc nhận dạng thức ăn sẽ giúp chúng ta nhận diện đầy đủ và chính xác các món ăn truyền thống của Việt Nam.

Trong khuôn khổ nghiên cứu này, chúng tôi sẽ áp dụng các mô hình Yolov7, SSD, RetinaNet để nhận dạng thức ăn truyền thống Việt Nam. Chúng tôi hy vọng cung cấp các kết quả thực nghiệm chính xác và hữu ích cho cộng đồng khoa học và các nhà phát triển công nghệ. Các mô hình này sẽ được áp dụng để

tạo ra một dataset để đánh giá các mô hình one-stage và sử dụng cho việc nhận diện các món ăn real-time. Bằng cách sử dụng mô hình one-stage, chúng tôi có thể dự đoán nhiều món ăn cùng lúc trên một khung hình. Chúng tôi lựa chọn nghiên cứu về nhận dạng thức ăn truyền thống Việt Nam sử dụng học sâu vì việc quản lý và đánh giá chất lượng của các món ăn truyền thống cần đến một phương pháp mới và hiệu quả hơn. Ngoài ra, nhận định thức ăn truyền thống Việt Nam là một phần không thể tách rời của văn hoá đất nước, việc nhận dạng và đánh giá chất lượng của các món ăn truyền thống sẽ góp phần bảo tồn và phát triển văn hoá ăn uống Việt Nam. Với sự phát triển của công nghệ học sâu, việc sử dụng các mô hình Yolov7, SSD và RetinaNet để nhận dạng thức ăn sẽ giúp cho việc đánh giá chất lượng một cách nhanh và chính xác hơn. Nghiên cứu này sẽ đóng góp cho việc xây dựng một dataset cho việc đánh giá các mô hình one-stage, đưa ra ý tưởng sử dụng model one-stage cho việc nhận diện các món ăn thời gian thực và sẽ dự đoán được nhiều món ăn cùng lúc trên một khung hình. Tóm lại, việc nhận dạng thức ăn truyền thống Việt Nam sử dụng học sâu là một bước tiến đột phá trong việc sử dụng công nghệ để giúp cho cuộc sống của chúng ta trở nên dễ dàng hơn và tốt hơn.

2 VNTF22 dataset

2.1 Thu thập và gán nhãn dữ liệu

Thu thập dữ liệu Dữ liệu được thu thập thủ công bằng cách tìm kiếm và tải về từ Google Image hoặc các ảnh chụp màn hình từ các video trên mạng xã hội Facebook, Youtube, Tiktok. Chúng tôi tiến hành thu thập dữ liệu về 16 món ăn truyền thống của Việt Nam bao gồm: bánh bò, bánh bột lọc, bánh chưng, bánh cam, bánh ít, bánh da lợn, bánh khọt, bánh cuốn, bánh tét, chả lụa, chè trôi nước, bánh mì, bánh xèo, gỏi cuốn, nem rán, phở. Dữ liệu được lưu trữ dưới định dạng đuôi '.png'.

Gán nhãn dữ liệu Để gán nhãn dữ liệu cho dự án này cần sử dụng một công cụ hỗ trợ để vẽ vùng bọc quanh đối tượng muốn nhận diện và gán tên cho đối tượng đó. Chúng tôi đã sử dụng trang web makesense.ai để thực hiện công việc này.

Quy trình gán nhãn dữ liệu cho mô hình bao gồm các bước sau:

- Nghiên cứu định dạng gán nhãn của từng mô hình.
- Nghiên cứu cách sử dụng makesense.ai để hỗ trợ gán nhãn.
- Vẽ vùng bọc: Sử dụng trang web hỗ trợ để vẽ vùng bọc quanh đối tượng muốn nhận diện trong từng hình ảnh.
- Gán tên: Gán tên cho đối tượng vừa vẽ vùng bọc.
- Lưu dữ liệu nhãn: Lưu các vùng bọc và tên đối tượng vừa gán vào một tập tin dữ liệu đầu ở dạnh xml cho mô hình SSD và RetinaNet và định dạng yolo cho mô hình Yolov7.

Việc gán nhãn dữ liệu là một quá trình tốn thời gian và yêu cầu độ chính xác cao. Vì chúng ảnh hưởng đến khả năng nhận diện vật thể và khả năng hội tụ về đúng vùng bọc đã vẽ của các vật thể ở các mô hình.

2.2 Tăng cường dữ liệu

Việc thu thập dữ liệu phụ vụ cho dự án có thể tốn rất nhiều thời gian. Vì vậy, nhằm giải quyết vấn đề này khi phải thu thập hàng nghìn hình ảnh đào tạo, tăng cường hình ảnh (data augmentation)[4] đã được phát triển để tạo dữ liệu đào tạo từ một tập dữ liệu hiện có. Tăng cường hình ảnh là quá trình sử dụng các hình ảnh có trong tập dữ liệu đã thu thập từ trước và tiến hành thao tác các kĩ thuật xử lý khác nhau với chúng để tạo ra nhiều phiên bản thay đổi của cùng một hình ảnh. Điều này vừa cung cấp nhiều hình ảnh hơn để huấn luyện mô hình, đồng thời vừa cho phép mô hình được tiếp xúc với đa dạng tình huống, bối cảnh từ ánh sáng, màu sắc và những chất lượng khác nhau trên cùng một tấm ảnh.

Trong dự án này, chúng tôi đã nghiên cứu các kĩ thuật làm nhiễu (Noise), làm mờ (Blur) cùng với xử lí pixel (Pixelate) trên hình ảnh, đồng thời kết hợp với đó là thay đổi độ sáng (Intensity) của ảnh:

- Gaussian Blur[6] là một thuật toán làm mờ hình ảnh sử dụng phép tính trung bình trong OpenCV. Nó sử dụng một ma trận trung bình Gaussian để làm mờ hình ảnh và giảm nhiễu. Kết quả của thuật toán này là một hình ảnh có độ mờ tùy chọn và khả năng giảm nhiễu tốt hơn.

- Gaussian Noise là một loại nhiễu được tạo bởi việc thêm một số ngẫu nhiên theo phân phối Gaussian vào hình ảnh gốc. Từ đó, kỹ thuật này có thể tạo ra những hình ảnh bị nhiễu theo nhiều mức độ khác nhau tùy thuộc vào thông số được điều chỉnh từ trước.

- Pixelate image là một kỹ thuật giảm độ phân giải của hình ảnh bằng cách chia nó thành nhiều ô hình vuông và thay đổi giá trị màu của mỗi ô để tạo ra một hình ảnh gồm các điểm lớn. Trong dự án này, chúng tôi đã pixelate hình ảnh bằng cách sử dụng các hàm resize và tính toán màu sắc trung bình của các ô.

- Intensity image là kỹ thuật thay đổi độ sáng hoặc độ tối của ảnh. Trong các tấm ảnh, độ sáng được biểu diễn bằng những giá trị từ (đen) đến 255 (trắng). Để thay đổi độ sáng của hình ảnh chúng tôi đã dựa vào tính chất này của ảnh để tăng hoặc giảm độ sáng của chúng.

2.3 VNTF22

Lấy cảm hứng từ bộ dữ liệu VinaFood21[5], Vietnam Traditional Food 2022 (VNTF22) là bộ dữ liệu hình ảnh về các món ăn truyền thống của Việt Nam. Sau khi được thu thập bộ dữ liệu bao gồm các hình ảnh nguyên bản của dữ liệu được tải về và các hình ảnh được tăng cường thông qua các filter. Bộ dữ liệu có tổng cộng 4742 hình ảnh các món ăn truyền thống Việt Nam bao gồm 16 món ăn. Số lượng ảnh của từng món ăn được thống kê tại bảng 1.

Bảng 1. Thống kê số lượng ảnh của món ăn trong bộ dữ liệu

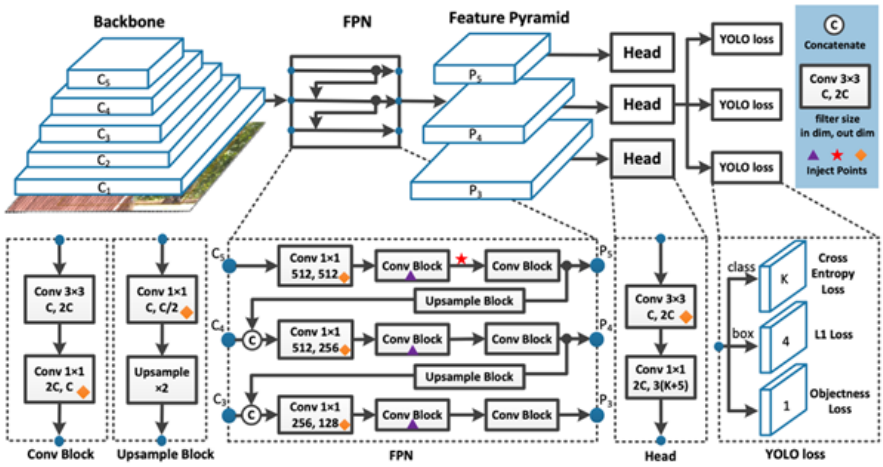
Tên món ăn	Số lượng ảnh nguyên bản	Số lượng ảnh tăng cường	Tổng số lượng ảnh
Bánh bò	149	149	298
Bánh bột lọc	149	149	298
Bánh cam	149	149	298
Bánh chưng	150	150	300
Bánh cuốn	149	149	298
Bánh da lợn	150	150	300
Bánh ít	150	150	300
Bánh khọt	149	149	298
Bánh mì	139	139	278
Bánh tét	140	140	280
Bánh xèo	148	148	296
Chả lụa	150	150	300
Chè trôi nước	150	150	300
Gỏi cuốn	149	149	298
Nem rán	150	150	300
Phở	150	150	300

3 Phương pháp

3.1 Mô hình one-stage cho bài toán object detection

YOLOv7 model YOLOv7[2] là một phương pháp nhận dạng đối tượng một bước sử dụng mạng neural để dự đoán vị trí và nhãn của các đối tượng trong hình ảnh. Nó được thiết kế để cải thiện tốc độ và chính xác so với các phương pháp nhận dạng đối tượng hai bước truyền thống như Faster R-CNN và RetinaNet. Cấu trúc của YOLOv7 bao gồm một mạng neural cơ sở (backbone) và một mạng neural đầu (head). Mạng neural cơ sở là một mạng đã được pre-train, như ResNet hoặc FPN, được sử dụng để trích xuất đặc trưng từ hình ảnh. Mạng neural đầu là một số lớp convolution và upsampling được áp dụng lên các bản đồ đặc trưng được trích xuất từ mạng neural cơ sở để dự đoán vị trí và nhãn của các đối tượng.

Phương pháp YOLOv7 sử dụng một grid để chia hình ảnh thành các ô vuông và dự đoán xác suất của mỗi đối tượng trong mỗi ô vuông. Mỗi ô vuông sẽ có một số lượng tùy chỉnh các anchor boxes để dự đoán vị trí của các đối tượng. YOLOv7 sử dụng thuật toán dự đoán số lượng đối tượng (objectness) và dự đoán vị trí của các đối tượng (location prediction) để tìm ra các đối tượng có thể trong hình ảnh. Một lợi thế của YOLOv7 là sử dụng cải tiến trong việc sử dụng anchor boxes, giúp cho việc dự đoán vị trí của các đối tượng trở nên chính xác hơn. YOLOv7 cũng sử dụng một số phương pháp cải tiến khác như sử dụng một mạng neural cơ sở mới và sử dụng cải tiến trong việc học bằng gradient. YOLOv7 được huấn luyện với một hàm mất mát tổng hợp bao gồm hàm mất mát cross-entropy cho phân loại và hàm mất mát L1 cho việc dự đoán vị trí của đối tượng.



Hình 1. Kiến trúc mô hình Yolov7

Công thức toán học chính của YOLOv7 được sử dụng để tính xác suất của một đối tượng xuất hiện trong một ô vuông được cho là:

$$P(object) = 1 - P(noobject) \tag{1}$$

Trong đó, $P(object)$ là xác suất của một đối tượng xuất hiện trong ô vuông đó, và $P(noobject)$ là xác suất của một đối tượng không xuất hiện trong ô vuông đó. Để dự đoán vị trí của đối tượng, YOLOv7 sử dụng một số anchor boxes được định nghĩa trước.

Mỗi anchor box được biểu diễn bằng 4 tham số (x, y, w, h) biểu thị vị trí và kích thước của anchor box.

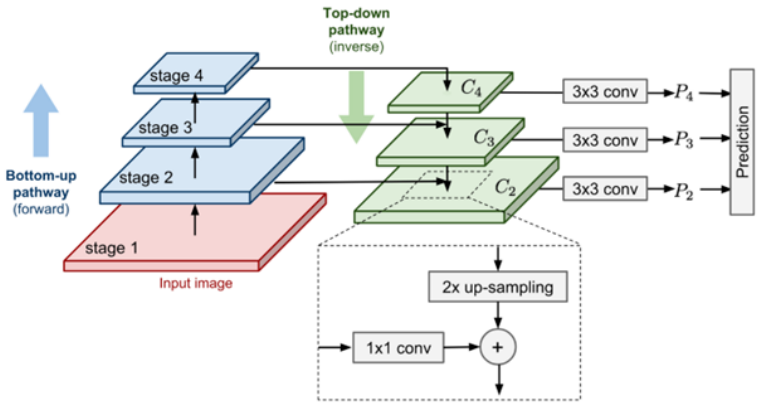
Công thức toán học để dự đoán vị trí của đối tượng trong một ô vuông và trong một anchor box được cho là:

$$(tx, ty, tw, th) = (x, y, w, h) + (dx, dy, dw, dh) \tag{2}$$

Trong đó, (x, y, w, h) là vị trí và kích thước của anchor box, và (dx, dy, dw, dh) là biến đổi cần thiết để chuyển đổi từ anchor box sang vị trí thực sự của đối tượng.

RetinaNET model RetinaNet là một kiểu mô hình object detection được sử dụng để dự đoán vị trí của đối tượng trong hình ảnh. Nó được xây dựng dựa trên mô hình FPN (Feature Pyramid Network) và sử dụng một cấu trúc anchor-based để dự đoán vị trí của đối tượng.

Trong kiến trúc RetinaNet[3], một mạng CNN được sử dụng làm backend, chuyển đổi hình ảnh vào các feature map, sau đó FPN được sử dụng để tạo ra



Hình 2. Kiến trúc mô hình RetinaNet

các feature map cấp cao và cấp thấp từ feature map của backend. Mỗi feature map được sử dụng để dự đoán các anchor với khả năng là một đối tượng. Một mạng fully connected được sử dụng để dự đoán xác suất cho mỗi anchor là một đối tượng và offset cho vị trí của đối tượng.

Trong phương pháp dự đoán của RetinaNet, các anchor được chia thành hai loại là positive anchors và negative anchors. Positive anchors là các anchor có vị trí gần với một đối tượng thực tế và negative anchors là các anchor không có đối tượng. Huấn luyện mô hình được thực hiện bằng cách tối ưu hóa hàm loss Focal loss đối với positive anchors và negative anchors. Trong kiến trúc RetinaNet, một FPN được sử dụng để tạo ra các feature map cấp cao và cấp thấp từ một backbone CNN. Mỗi feature map được sử dụng để dự đoán các anchor với khả năng là một đối tượng.

Một mạng fully connected được sử dụng để dự đoán xác suất cho mỗi anchor là một đối tượng và offset cho vị trí của đối tượng.

Phương thức chính được sử dụng trong RetinaNet là huấn luyện mô hình bằng cách sử dụng hàm loss Focal loss, được định nghĩa như sau:

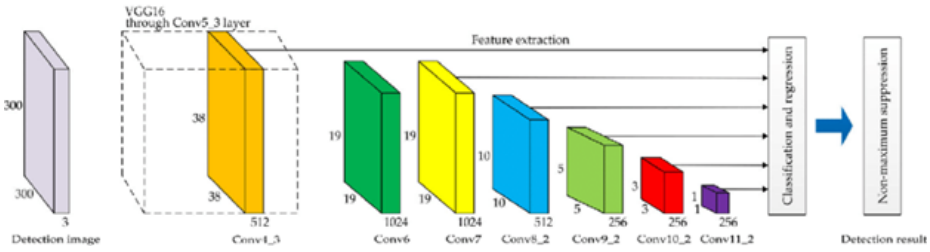
$$Focalloss = -(1 - pt)^{\gamma} * \log pt \tag{3}$$

Trong đó:

- pt là xác suất dự đoán đúng của mô hình cho mỗi điểm dữ liệu.
- γ là một hệ số tùy chỉnh được sử dụng để tăng cường sự quan tâm của mô hình đến các điểm dữ liệu có xác suất thấp hơn.

SSD model Single Shot MultiBox Detector (SSD)[1] là một phương pháp để phát hiện đối tượng trong hình ảnh. Nó sử dụng một mô hình neural network để dự đoán các bounding box và class scores cho các đối tượng trong hình ảnh. Cấu trúc của SSD bao gồm một base network, ví dụ như VGG hoặc ResNet,

được sử dụng để trích xuất đặc trưng từ hình ảnh đầu vào. Sau đó, một số lượng các tầng mở rộng được thêm vào để tăng cường đặc trưng và giúp cho việc dự đoán bounding box trở nên chính xác hơn. Các tầng mở rộng này bao gồm các tầng convolutional và tầng fully connected. Một trong những điểm mạnh của SSD là nó sử dụng một số lượng lớn các anchor boxes để dự đoán các bounding box cho các đối tượng. Anchor boxes là các khung hình được dự đoán trước đó trong hình ảnh và được sử dụng như một gợi ý cho việc dự đoán bounding box thực sự của đối tượng.



Hình 3. Kiến trúc mô hình SSD

SSD cũng sử dụng một phương pháp gọi là hard negative mining để giảm thiểu việc quá hạn hoạt động của các anchor boxes không tương ứng với bất kỳ đối tượng nào trong hình ảnh. Trong phương pháp này, những anchor boxes không tương ứng với bất kỳ đối tượng nào trong hình ảnh được gọi là negative examples, và chúng được chọn ra để huấn luyện mô hình trong suốt quá trình huấn luyện. Điều này giúp cho việc dự đoán bounding box trở nên chính xác hơn và giảm thiểu việc quá hạn hoạt động của các anchor boxes không tương ứng.

Trong SSD, có một số công thức toán học được sử dụng để tính toán các bounding box và class scores của các đối tượng trong hình ảnh:

- Công thức tính toán Non-Maximum Suppression (NMS) để loại bỏ các bounding box trùng lặp: đây là một phương pháp xử lý được sử dụng sau khi dự đoán bounding box để loại bỏ các bounding box có độ tương đồng cao với nhau. Các công thức trên được sử dụng để huấn luyện và sử dụng mô hình SSD để dự đoán bounding box và class scores cho các đối tượng trong hình ảnh.

- Công thức tính toán Intersection over Union (IoU) giữa một bounding box dự đoán và một bounding box thật sự của một đối tượng: đây là một chỉ số đo lường độ tương đồng giữa hai bounding box.

+ Intersection over Union (IoU)[8] giữa hai bounding box A và B được tính bằng công thức:

$$IoU_{(A,B)} = (A \cap B) \,/\, (A \cup B) \tag{4}$$

- Công thức tính toán MultiBox loss: đây là một hàm mất mát được sử dụng để huấn luyện mô hình SSD. Nó bao gồm hai phần chính: hàm mất mát cho việc

dự đoán bounding box và hàm mất mát cho việc dự đoán class scores. MultiBox loss gồm hai phần chính:

+ Hàm mất mát cho việc dự đoán bounding box: được tính bằng công thức hàm mất mát smooth L1

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, if\ |x| < 1 \\ |x| - 0.5, otherwise \end{cases} \tag{5}$$

+ Hàm mất mát cho việc dự đoán class scores: được tính bằng công thức hàm mất mát Cross-Entropy

$$L = -\sum y_{true} * \log y_{pred} \tag{6}$$

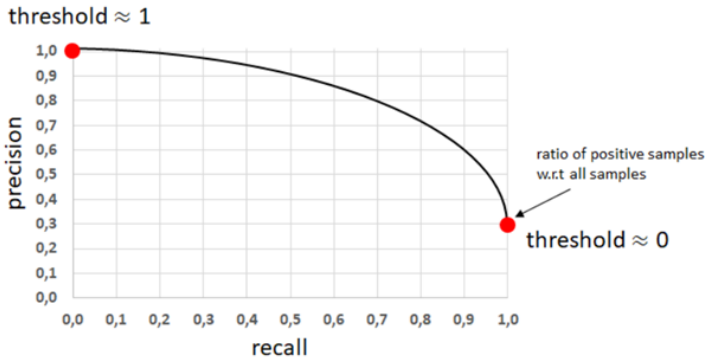
3.2 Độ đo (mAP)

Độ đo mAP (Mean Average Precision)[7] là trung bình cộng giá trị AP của các class khác nhau. Ở đây chúng tôi giải quyết bài toán Object Detection có nhiều class, mỗi class ta sẽ tiến hành đo AP (Average Precision), lấy trung bình của tất cả các giá trị AP của các class, thu được chỉ số mAP của mô hình.

$$mAP = \left(\sum_{i=1}^{n} AP_i\right)/n \tag{7}$$

Với: n là số class cần nhận diện

Để thu được đươc chỉ số AP, ta cần biết các chỉ số Precision, Recall, IoU (Intersection over Union). Dựa trên quan hệ giữa Precison và Recall khi IoU thay đổi, ta có được Precision Recall Curve. AP chính là phần diện tích nằm dưới Precision Recall Curve.



Hình 4. Hình ảnh minh họa Precision Recall Curve

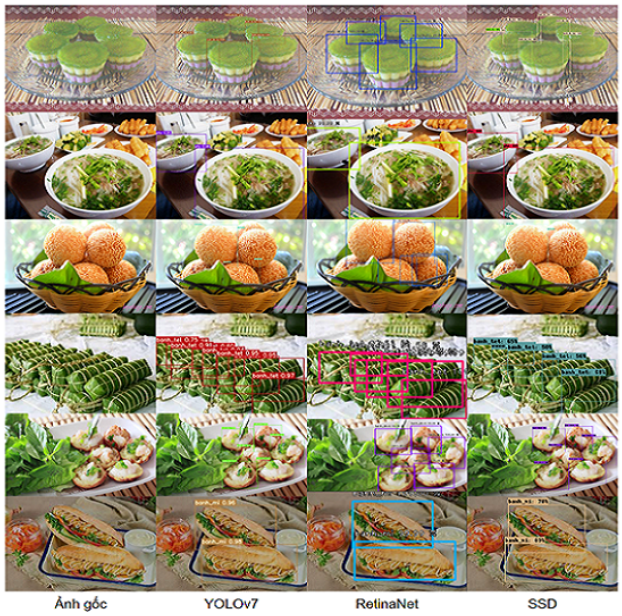
4 Kết quả & thảo luận

Sau khi huấn luyện bộ dữ liệu VNTF22 cho các mô hình one-stage, kết quả thử nghiệm được tính toán theo độ đo Average Precision (AP) trung bình trên 16 lớp và các thông số khác đã được ghi lại trong Bảng 2.

Bảng 2. Bảng so sánh kết quả đánh giá trên tập dữ liệu thử nghiệm

	#Param.	Size	AP^{Val}	AP^{Val}_{50}	AP^{Val}_{75}
YOLOv7	36M	640	81.5%	92.7%	92.3%
RetinaNet – Resnet50	36M	800	71.9%	87.0%	78.5%
SSD – Mobilenet v2	15M	320	70.4%	89.0%	82.3%

Sự so sánh giữa các mô hình cho thấy rằng, mô hình YOLOv7 được huấn luyện trên tập dữ liệu đánh giá cho kết quả tốt nhất với mAP là 81.5%. Điều này cho thấy rằng, mô hình YOLOv7 học khá tốt với bộ dữ liệu VNTF22 này. Tuy nhiên, hai mô hình còn lại, RetinaNet và SSD, cũng cho kết quả tương đương nhau, với mAP cho RetinaNet là 71.9% và cho SSD là 70.4%.

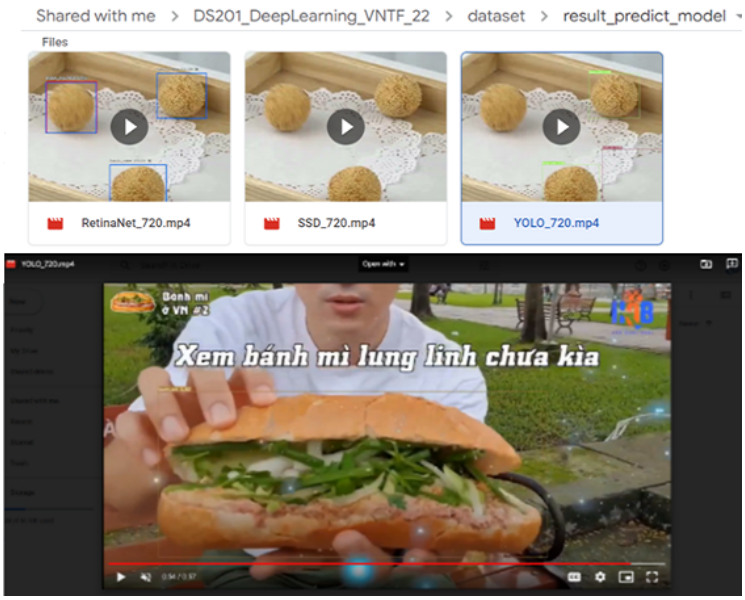


Hình 5. Hình ảnh kết quả dự đoán của mô hình trên dữ liệu ngẫu nhiên

Nhận xét về các kết quả trên cho thấy rằng, mô hình SSD có số lượng tham số ít hơn (chỉ 15M) so với mô hình RetinaNet (36M), nhưng vẫn cho kết quả mAP gần bằng với mô hình RetinaNet. Điều này có thể cho thấy rằng, mô hình SSD có thể là một lựa chọn tốt hơn khi cần sử dụng trong các hệ thống cần tối ưu về tài nguyên.

Tuy nhiên, để đưa ra một quyết định chính xác về mô hình nào sẽ sử dụng, cần có thêm nghiên cứu về độ chính xác của các mô hình trong các tình huống thực tế và đánh giá về hiệu năng của mô hình YOLOv7.

Trong việc đánh giá hiệu năng của các mô hình, mô hình YOLOv7 đã cho thấy hiệu năng tốt nhất khi được huấn luyện trên tập dữ liệu VNTF22. Kết quả cho thấy, mô hình YOLOv7 có độ đo Average Precision (mAP) trung bình trên 16 class là 81.5%. Điều này cho thấy mô hình YOLOv7 đã học tốt với bộ dữ liệu được sử dụng trong quá trình huấn luyện. Tuy nhiên, nếu so sánh với mô hình RetinaNet với mAP là 71.9% và mô hình SSD có mAP là 70.4%, mô hình YOLOv7 vẫn còn cần được cải tiến thêm để có thể đạt được hiệu năng tốt hơn.



Hình 6. Kết quả dự đoán của mô hình one-stage trên video

5 Kết luận

Trong bài báo cáo này, chúng tôi đã giới thiệu một bộ dữ liệu mới về các món ăn truyền thống của người Việt Nam bao gồm 16 món ăn khác nhau, VNTF22.

Bộ dữ liệu có gần 4800 ảnh các món ăn đã được chúng tôi tăng cường dữ liệu từ 1600 tấm ảnh thu thập trên Google. VNTF22 đã được chúng tôi gán nhãn theo 2 định dạng file xml và txt cho việc sử dụng để đánh giá các mô hình one-stage. Ngoài ra, nhóm cũng đưa ra ý tưởng sử dụng model one-stage cho việc nhận diện các món ăn real-time và có thể dự đoán nhiều món ăn cùng lúc trên 1 khung hình. Các mô hình one-stage như YOLO, RetinaNet, SSD được chúng tôi huấn luyện thử nghiệm trên bộ dữ liệu và YOLOv7 đạt được mAP tốt nhất trên 81% trên tập test mà bộ dữ liệu cung cấp.

Trong quá trình thu thập, chúng tôi cũng gặp một số hạn chế ảnh hưởng đến chất lượng của bộ dữ liệu như khó tìm kiếm các nguồn ảnh có chất lượng tốt, ảnh bị lặp và trùng nhau, ảnh bị mờ, nhiễu,...

Trong tương lai, bộ dữ liệu có thể được thu thập thêm để tăng thêm sự đa dạng của dữ liệu và mở rộng thêm nhiều thức ăn truyền thống của Việt Nam khác. Thực hiện tăng cường dữ liệu bằng nhiều phương pháp khác. Thử nghiệm huấn luyện trên nhiều mô hình hơn để đánh giá mức độ hiệu quả của bộ dữ liệu và triển khai các mô hình đã huấn luyện đó lên các thiết bị di động để tiến hành thử nghiệm thử trong thực tế.

Chúng tôi hy vọng với bộ dữ liệu VNTF22 có thể đóng góp cho cộng đồng nghiên cứu nhằm mục đinh giải quyết các bài toán về Vietnamese food recognition và đánh giá các mô hình one-stage.

Tài liệu

1. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of big data **6**(1), 1–48 (2019)
2. Gedraite, E.S., Hadad, M.: Investigation on the effect of a gaussian blur in image filtering and segmentation, 393–396 (2011). IEEE
3. Nguyen, T.T., Nguyen, T.Q., Vo, D., Nguyen, V., Ho, N., Vo, N.D., Van Nguyen, K., Nguyen, K.: Vinafood21: A novel dataset for evaluating vietnamese food recognition, 1–6 (2021). IEEE
4. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: Yolov7: Trainable bag-of- freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696 (2022)
5. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection, 2980–2988 (2017)
6. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector, 21–37 (2016). Springer
7. ezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 658–666 (2019)
8. Henderson, P., Ferrari, V.: End-to-end training of object class detectors for mean average precision. In: Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13, pp. 198–213 (2017). Springer