Đếm số người và phát hiện vi phạm khoảng cách cộng đồng từ CCTV

Đỗ Văn Nam[1], Lê Đình Bảo Long[2], Trần Nhật Nam[3], and Trần Thành Luân[4]

[1] University of Information Technology, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
{19521866,19521872, 19521782, 19521810}@gm.uit.edu.vn

Abstract. Hiện nay tình hình dịch bệnh đang căng thẳng trên toàn thế giới, nhiều chính phủ đã ban hành việc cấm tập trung đông người cũng như là giãn cách xã hội ở những nơi có nguy cơ cao như nhà hàng, phòng hội họp,... Trong bài báo cáo này chúng tôi sẽ trình bày phương pháp để đếm số lượng người và phát hiện khoảng cách xã hội bằng cách sử dụng học sâu để đánh giá khoảng cách giữa mọi người nhằm giảm thiểu tác động của đại dịch coronavirus này. Công cụ phát hiện được phát triển để cảnh báo mọi người duy trì khoảng cách an toàn với nhau bằng cách đánh giá nguồn cấp dữ liệu video. Khung video từ máy ảnh được sử dụng làm đầu vào và mô hình phát hiện đối tượng nguồn mở được đào tạo trước dựa trên thuật toán YOLOv4 được sử dụng để phát hiện người có mặt trong khung hình. Sau đó, khung hình video được chuyển thành chế độ xem từ trên xuống để đo khoảng cách từ mặt phẳng 2D bằng thuật toán Chebyshev. Khoảng cách giữa mọi người có thể được ước tính và bất kỳ cặp người nào không tuân thủ trong màn hình sẽ được biểu thị bằng khung màu đỏ và đường màu đỏ. Phương pháp đề xuất đã được xác nhận trên một đoạn video quay trước những người trong 1 sân bay. Kết quả cho thấy rằng phương pháp được đề xuất có thể xác định các thước đo khoảng cách xã hội giữa nhiều người trong video. Kỹ thuật đã phát triển có thể được phát triển thêm như một công cụ phát hiện trong ứng dụng thời gian thực.

Keywords: Object Detection · Distance social · Deeplearning · YOLOv4.

1 Giới thiệu

Bài toán của chúng tôi đưa ra sẽ là phát hiện số người từ frame của camera an ninh từ đó đếm số lượng người có mặt bằng cách sử dụng YOLOv4. Sau khi đã có các đối tượng thì thực hiện đo khoảng cách giữa các đối tượng để chỉ những vị phạm về khoảng cách.

Nhận diện vật thể (Object Recognition) là 1 thuật ngữ chung dùng để chỉ các bài toán thuộc thị giác máy tính (Computer Vision) có liên quan tới việc nhận diện các vật thể trong ảnh kỹ thuật số. Ta có thể phân nó thành 3 loại chính:

– **Phân loại ảnh(Image Classification)**: Dự đoán ảnh sẽ thuộc lớp nào trong số các lớp đã cho trước.
  • Input: 1 bức ảnh có một vật thể duy nhất.
  • Output: Lớp của bức ảnh trên.
– **Định vị vật thể (Object Localization)**: Định vị vị trí của các vật thể trong ảnh và chỉ ra bằng cách vẽ bounding box xung quanh vật thể đó.
  • Input: 1 bức ảnh với một hoặc nhiều vật thể.
  • Output: 1 hoặc nhiều bounding box.
– **Phát hiện vật thể (Object Detection)**: Giống như Định vị vật thể nhưng bounding box sẽ đi kèm tên lớp của vật thể đó.
  • Input: 1 bức ảnh với một hoặc nhiều vật thể.
  • Output: 1 hoặc nhiều bounding box đi kèm với tên class ứng với mỗi bounding box.

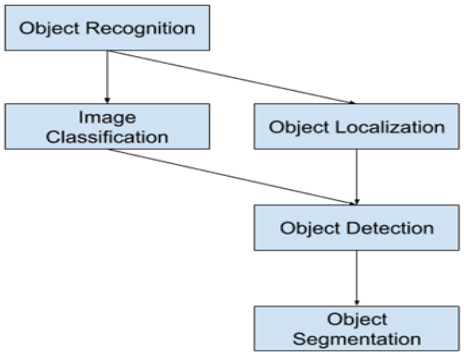

Fig. 1. Các loại bài toán chính trong Object Recognition

Ở đồ án lần này, chúng tôi sẽ tập trung vào bài toán Phát hiện vật thể (Object Detection), cụ thể là phát hiện đầu người. Trong thời điểm đại dịch toàn cầu Covid-19 đã bước sang năm thứ 2 và vẫn chưa có dấu hiệu suy giảm nhiều về số ca nhiễm, việc tuân thủ các quy định về giãn cách nơi công cộng là rất cần thiết. Tuy nhiên, không phải ai cũng có ý thức chấp hành tốt, việc phát hiện các vi phạm này một cách tự động sẽ giúp tiết kiệm nhân lực cũng như thời gian. Chính vì vậy, mục tiêu của đồ án lần này là phát hiện số người có trong 1 khu vực nhỏ như xe bus, sảnh chờ....Từ đó đưa ra các phương án xử lí hiệu quả.

Bài toán nhỏ thứ 2 sẽ áp dụng kết quả phát hiện đầu người thu được từ bài toán trên sau đó sẽ đưa từng khung hình trên video thành dạng hình ảnh 2D và đo khoảng cách các đối tượng bằng một thuật toán toán nào đó. Mà trong bài báo cáo này chúng tôi lựa chọn thuật toán tính khoảng cách Chebyshev.

Để hoàn thành mục tiêu đã nêu ra, chúng tôi sử dụng yolov4 – mô hình pretrained trên bộ dữ liệu MS COCO với file weight yolov4.weight.

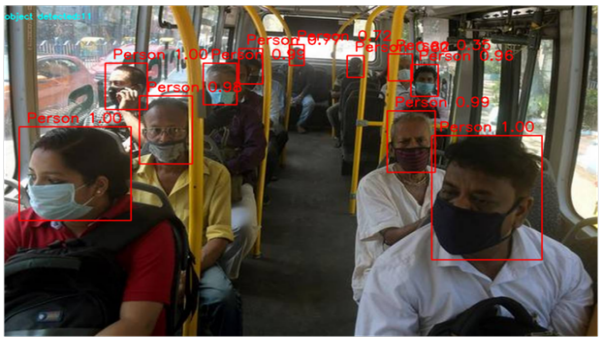

Fig. 2. Phát hiện đầu người trên xe bus

2 Bộ dữ liệu

2.1 Thu thập dữ liệu

Trong đồ án này, tập dữ liệu dùng để nhận dạng người được thu thập bằng hai cách đó là lấy một phần bộ dữ liệu có sẵn và tự thu thập:

- Đối với hướng đầu tiên, thu thập dữ liệu có sẵn, việc thu thập dữ liệu sẽ nhanh và hiệu quả hơn hướng thứ hai. Cụ thể ở phần này, chúng tôi đã trích ra 2000 ảnh trong Dataset Scut Head.
- Đối với hướng thứ hai, thu thập dữ liệu bằng cách thủ công, việc thu thập dữ liệu sẽ bổ sung được những dữ liệu khó nhận dạng như ảnh mờ, trời mưa,... Cụ thể ở phần này, chúng tôi đã thu thập được 2329 ảnh trên các wedsite thông dụng như Google, Bing,...

2.2 Khó khăn trong việc gán nhãn dữ liệu

Trong quá trình gán nhãn dữ liệu thì có 1 số thách thức mà chúng tôi gặp phải như là:

- Ảnh mờ.
- Ảnh trong điều kiện thiếu sáng.
- Có những poster quảng cáo chứa hình người trong ảnh.

→ Để giải quyết những khó khăn này thì chung tôi sẽ tiến hành cho 1 thành viên gán nhãn và 3 thành viên xem xét lại để tránh những sai sót, và không gán nhãn những người có trong poster quảng cáo.

2.3 Tăng cường dữ liệu

Sau khi thu thập dữ liệu và thực nghiệm bằng việc train mô hình thì chúng tôi nhận thấy rằng dữ liệu dùng để train model là khá ít, việc này sẽ rất khó để tạo

ra được kết quả tốt trong dự đoán. Do đó, chúng tôi cần đến một kỹ thuật gọi là tăng cường dữ liệu (Data augmentation) để tạo ra được nhiều dữ liệu hơn từ những dữ liệu đã có sẵn. Sau khi tìm hiểu các tool dùng để tăng cường dữ liệu thì chúng tôi chọn tool trên Roboflow.

Bộ dữ liệu được tăng cường như sau:

- Bounding Box Crop: 25% Minimum Zoom, 50% Maximum Zoom.
- Bounding Box Blur: Làm mờ đến tối đa 7px.

Sau khi tăng cường dữ lệu, bộ dữ liệu của chúng tôi đã tăng thêm 300% so với dữ liệu ban đầu.

3 Nội dung

3.1 Bài toán phát hiện đầu người

Chúng tôi lựa chọn mô hình mạng nổi tiếng trong bài toán hiện này đó là mô hình mạng YOLO, cùng tìm hiểu kĩ lý thuyết về mô hình mạng học sâu này.

YOLO Object Detection[1] là một bài toán quan trọng trong lĩnh vực Computer Vision, thuật toán Object Detection được chia thành 2 nhóm chính:

- Họ các mô hình RCNN (Region-Based Convolutional Neural Networks) để giải quyết các bài toán về định vị và nhận diện vật thể.
- Họ các mô hình về YOLO (You Only Look Once) dùng để nhận dạng đối tượng được thiết kế để nhận diện các vật thể real-time.

YOLO[2] là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. YOLO được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers. Trong đó các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.

YOLOv4 YOLOv4[3] là một loạt các cải tiến về tốc độ so với YOLOv3 và được cài đặt từ một bản fork của Darknet. Kiến trúc của YOLOv4 đã đưa bài toán Object Detection dễ tiếp cận hơn với những người không có tài nguyên tính toán mạnh. Chúng ta hoàn toàn có thể huấn luyện một mạng phát hiện vật với độ chính xác rất cao bằng YOLOv4 chỉ với GPU 1080ti hoặc 2080ti. Trong tương lai, việc tối ưu lại các mạng hiện tại để phù hợp với tài nguyên tính toán yếu hoặc tạo ra sự song song hóa cao ở các server chắc chắn phải được thực hiện để có thể đưa các ứng dụng computer vision vào thực tế.

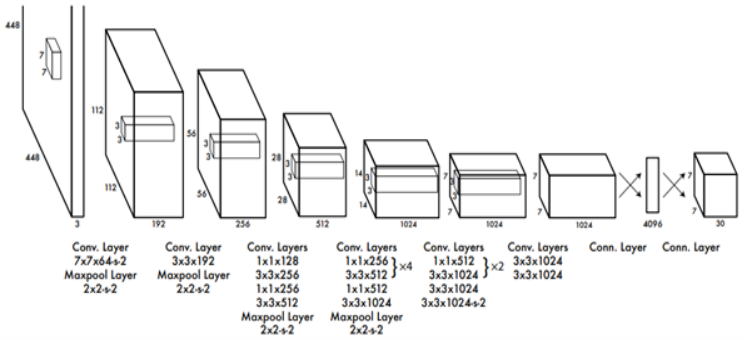

Fig. 3. YOLO

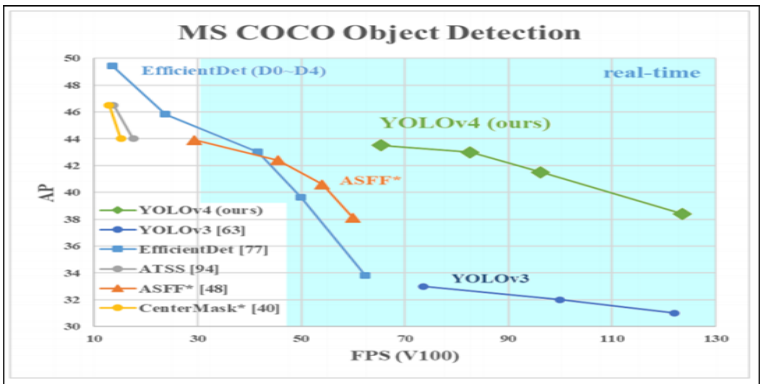

Fig. 4. So sánh Yolov4 với các mô hình phổ biến

Cách thức hoạt động Đầu vào của mô hình là một ảnh, mô hình sẽ nhận dạng ảnh đó có đối tượng nào hay không, sau đó sẽ xác định tọa độ của đối tượng trong bức ảnh. Ảnh đầu vào được chia thành S*S ô thường thì sẽ là 3*3, 7*7, 9*9... việc chia ô này có ảnh hưởng tới việc mô hình phát hiện đối tượng.

Với Input là 1 ảnh, đầu ra mô hình là một ma trận 3 chiều có kích thước S*S*(5*M+N) với số lượng tham số mỗi ô là (5×N+M) với N và M lần lượt là số lượng Box và Class mà mỗi ô cần dự đoán. Ví dụ với hình ảnh trên chia thành 7 ×7 ô, mỗi ô cần dự đoán 2 bounding box và 3 object : con chó, ô tô, xe đạp thì output là 7*7*13, mỗi ô sẽ có 13 tham số, kết quả trả về (7×7×2=98) bounding box. Chúng ta sẽ cùng giải thích con số 5×N+M) được tính như thế nào. Dự đoán mỗi bounding box gồm 5 thành phần : (x, y, w, h, prediction) với (x, y) là tọa độ tâm của bounding box, (w, h) lần lượt là chiều rộng và chiều cao của bounding box, prediction được định Pr(Object)× IOU(pred,truth) . Với hình ảnh trên như ta tính mỗi ô sẽ có 13 tham số, ta có thể hiểu đơn giản như sau tham số thứ 1 sẽ chỉ ra ô đó có chứa đối tượng nào hay không P(Object), tham số 2, 3, 4, 5 sẽ trả về x, y ,w, h của Box1. Tham số 6, 7, 8, 9, 10 tương tự sẽ Box2, tham số

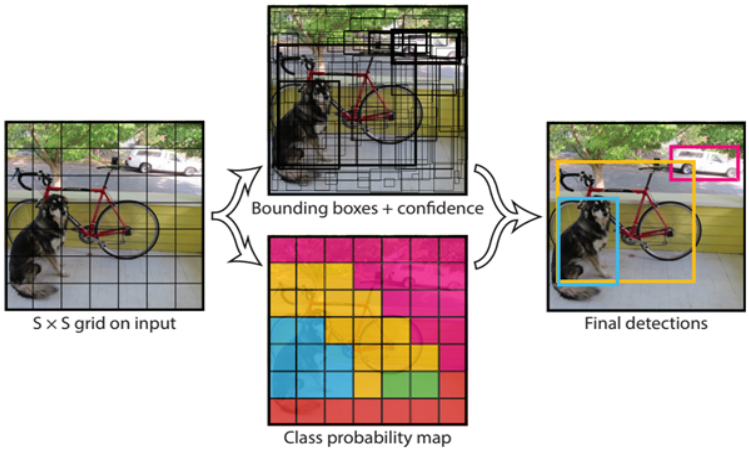

Fig. 5. Cách thức hoạt động của Yolo

11, 12, 13 lần lượt là xác suất ô đó có chứa object1(P(chó|object), object2(P(ô tô|object)), object3(P(xe đạp|object)). Lưu ý rằng tâm của bounding box nằm ở ô nào thì ô đó sẽ chứa đối tượng, cho dù đối tượng có thể ở các ô khác thì cũng sẽ trả về là 0. Vì vậy việc mà 1 ô chứa 2 hay nhiều tâm của bouding box hay đối tượng thì sẽ không thể detect được, đó là một hạn chế của mô hình YOLO1, vậy ta cần phải tăng số lượng ô chia trong 1 ảnh lên đó là lí do vì sao mình nói việc chia ô có thể làm ảnh hưởng tới việc mô hình phát hiện đối tượng.

Hàm tính IOU Trên ta có đề cập prediction được định nghĩa Pr(Object)× IOU(pred,truth), ta sẽ làm rõ hơn IOU(pred, truth) là gì. IOU (INTERSECTION OVER UNION) là hàm đánh giá độ chính xác của object detector trên tập dữ liệu cụ thể. IOU được tính bằng:

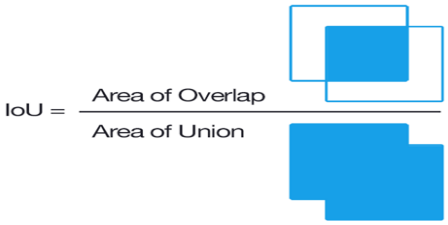

Fig. 6. Công thức tính IOU

Trong đó Area of Overlap là diện tích phần giao nhau giữa predicted bounding box với grouth-truth bouding box , còn Area of Union là diện tích phần hợp giữa predicted bounding box với grouth-truth bounding box. Những bounding box

được đánh nhãn bằng tay trong tập traing set và test set. Nếu IOU > 0.5 thì prediction được đánh giá là tốt.

Loss Function Hàm lỗi trong YOLO được tính trên việc dự đoán và nhãn mô hình để tính. Cụ thể hơn nó là tổng độ lỗi của 3 thành phần con sau:

- Độ lỗi của việc dự đoán loại nhãn của object - Classifycation loss.
- Độ lỗi của dự đoán tọa độ tâm, chiều dài, rộng của boundary box (x, y ,w, h) Localization loss.
- Độ lỗi của việc dự đoán bounding box đó chứa object so với nhãn thực tế tại ô vuông đó - Confidence loss.

3.2 Phát hiện vi phạm khoảng cách cộng đồng

Sau khi đã phát hiện từng đối tượng là đầu người trong 1 frame. Thì các frame đó được chuyển xuống chế độ xem từ trên xuống và được đưa vào trong không gian 2 chiều để tính khoảng cách giữa các đối tương.

Đề tính được khoảng cách này thì chúng tôi lựa chọn độ đo có tên là Chebyshev[4] để thực hiện cho bài toán của chúng tôi.

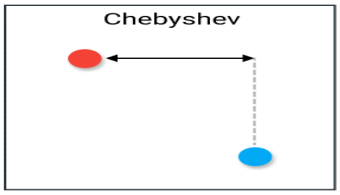

Fig. 7. Độ đo Chebyshev

$$D(x, y) = \max_i (|x_i - y_i|)$$

Chebyshev distance

Fig. 8. Công thức Chebyshev

Khoảng cách Chebyshev được định nghĩa là chênh lệch lớn nhất giữa hai vectơ dọc theo bất kỳ chiều tọa độ nào. Nói cách khác, nó chỉ đơn giản là khoảng cách lớn nhất dọc theo một trục. Do tính chất của nó, nó thường được gọi là khoảng cách Bàn cờ vì số nước đi tối thiểu của quân vua để đi từ ô vuông này sang ô khác bằng khoảng cách Chebyshev.

4 Mô hình và kết quả

4.1 Huấn luyện mô hình phát hiện đầu người

Sửa đổi thông số mô hình

– Chúng tôi sử dụng mô hình yolov4 pre-train để huấn luyện mô hình, Pre-train mô hình với việc chỉnh sửa các tham số trong file config cho việc nhận dạng người như sau:batch=64, subdivisions=32, class = 1, max_batches = 6000, step = 4800, 5400, filter = 18, with = 416, height = 416.
– Sử dụng file weight yolov4.conv.137 để pre-train. Việc huấn luyện mô hình được thực hiện trên google colab pro.

Chuẩn bị bộ dữ liệu Ở bài toán này, chúng tôi cho nó một class duy nhất để gán nhãn là person. Sau khi gán nhãn các hình ảnh, có tổng cộng 17209 hình ảnh, trong đó 17109 hình ảnh dành cho tập huấn luyện và 100 ảnh dành cho tập thử nghiệm.

Chạy mô hình Với max_batches = 6000 chúng tôi đã phải huấn luyện liên tục suốt 24 giờ. Trong khoảng thời gian đó chúng tôi đã lưu lại các mô hình tại các vòng 1000, 2000,...6000.

4.2 Thực nghiệm mô hình phát hiện đầu người

Sau khi hoàn thành công việc huấn luyện mô hình, chúng tôi tiến hành thực nghiệm mô hình trên các video trích xuất từ camera an ninh. Dưới đây là hình ảnh minh hoạ kết quả của chúng tôi:

Fig. 9. Mô hình phát hiện đầu người

Dựa và kết quả thực nghiệm ở trên ta có thể thấy mô hình đã nhận diện và đếm số người khá tốt. Một số người chúng ta có thể nhận biết bằng mắt thường nhưng không detect được đa số là bị mờ hoặc bị che khuất phần đầu.

Chúng tôi tiến hành đánh giá kết quả trên tập thử nghiệm với mô hình YOLOv4 khi chưa tăng cường dữ liệu và sau khi tăng cường dữ liệu, kết quả được đánh giá bằng bảng ở dưới:

Table 1. Bảng so sánh kết quả trước và sau khi tăng cường dữ liệu

	last_weight.weight cũ	**last_weight.weight mới**
precision	0.7	0.84
recall	0.58	0.92
F1-score	0.7	0.88
mAp	0.62	0.925
Average IoU	0.52	0.66

Kết quả cho thấy là trọng số mới cho kết quả rất tốt và khả quan, kết quả này hoàn toàn giải quyết được bài toán đầu tiên mà chúng tôi đặt ra.

4.3 Mô hình phát hiện vi phạm khoảng cách

[5] Sau khi đã có các frame hình từ mô hình trên chúng tôi tiến hành kết hợp cùng thư viện Distance được tích hợp sẵn trong thư viện python. Do kĩ thuật tìm khoảng cách trong ảnh còn rất khó khăn. Để đơn giản chúng tôi sẽ đưa frame về dạng hình ảnh trong không gian 2D để có thể sử dụng được thư viện distance trên.

Do độ đo trong hình ảnh là pixel nên lúc thiết lập thông số khoảng cách án toàn tối thiểu (MIN_DISTANCE) ta sẽ đổi khoảng cách tối thiểu về pixel. Trong quá trình thực nghiệm nhiều lần chúng tôi chọn được thông số MIN_DISTANCE tốt nhất là 75.

Sau đây là minh họa về kết quả chúng tôi thu được:

Fig. 10. Mô hình phát hiện vi phạm khoảng cách

Những người bị phát hiện không nằm trong vùng có khoảng cách an toàn sẽ được đánh ô màu đỏ và được cảnh bảo. Những người duy trì được khoảng cách an toàn sẽ được đánh dấu ô màu xanh.

5 Kết luận

5.1 Kết luận

Từ khoảng 5000 vòng trở đi, mô hình có xu hướng bị overfit dẫn đến kết quả trên tập thử nghiệm không được cao.

Kết quả thực nghiệm chứng minh được mô hình end-to-end cần nhiều dữ liệu hơn các mô hình khác nên việc tăng cường dữ liệu cho bộ dữ liệu này là rất cần thiết. Về công thức tính khoảng cách còn khá đơn giản và có thể chưa chính xác nhưng nó cũng là bước đầu trong việc nghiên cứu cũng như phát triển.

5.2 Hạn chế và cách khắc phục

Hạn chế: Mô hình cho kết quả không tốt với những đối tượng ở quá xa, bị che khuất hoặc quá nhiều đối tượng ở gần nhau.

Cách khắc phục: Tăng cường dữ liệu cho mô hình, khắc phục những điểm hạn chế như là bị mờ, bị che khuất.

Mô hình của chúng tôi có thể áp dụng thực tế trong bối cảnh tình hình dịch còn đang diễn biến khó khăn như vậy nhằm cảnh báo về việc duy trì khoảng cách xã hội.

References

1. Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
2. Tao, Jing, et al. "An object detection system based on YOLO in traffic scene." 2017 6th International Conference on Computer Science and Network Technology (ICCSNT). IEEE, 2017.
3. Jiang, Zicong, et al. "Real-time object detection method based on improved YOLOv4-tiny." arXiv preprint arXiv:2011.04244 (2020).
4. Rosser, J. Barkley, and Lowell Schoenfeld. "Sharper Bounds for the Chebyshev Functions." Mathematics of computation (1975): 243-269.
5. Bhambani, Krisha, Tanmay Jain, and Kavita A. Sultanpure. "Real-time Face Mask and Social Distancing Violation Detection System using YOLO." 2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC). IEEE, 2020.