TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ**

**NHẬP MÔN TRÍ TUỆ NHÂN TẠO**

*Người thực hiện*: **Trần Phi Khanh – 520H0648**

**Bùi Đức Dũng – 518H0611**

Lớp **: 20H50303**

**18H50301**

Khoá  **: 22-24**

*Người hướng dẫn*: **TS BÙI THANH HÙNG**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ**

**NHẬP MÔN TRÍ TUỆ NHÂN TẠO**

Người thực hiện: **Trần Phi Khanh**

**Bùi Đức Dũng**

Lớp **: 20H50303**

**18H50301**

Khoá  **: 22-24**

Người hướng dẫn: **TS. BÙI THANH HÙNG**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn thầy Bùi Thanh Hùng hướng dẫn nhiệt tình và đưa chúng em đồ án nghiên cứu đầy thú vị để hiểu thêm về trí tuệ nhân tạo. Qua đề tài mà thầy Bùi Thanh Hùng đã giao chúng em đã có cơ hội học hỏi thêm về thuật toán và hiểu được ứng dụng thiết thực của trí tuệ nhân tạo.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của TS. Bùi Thanh Hùng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Khanh*

*Dũng*

*Trần Phi Khanh*

*Bùi Đức Dũng*

**PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

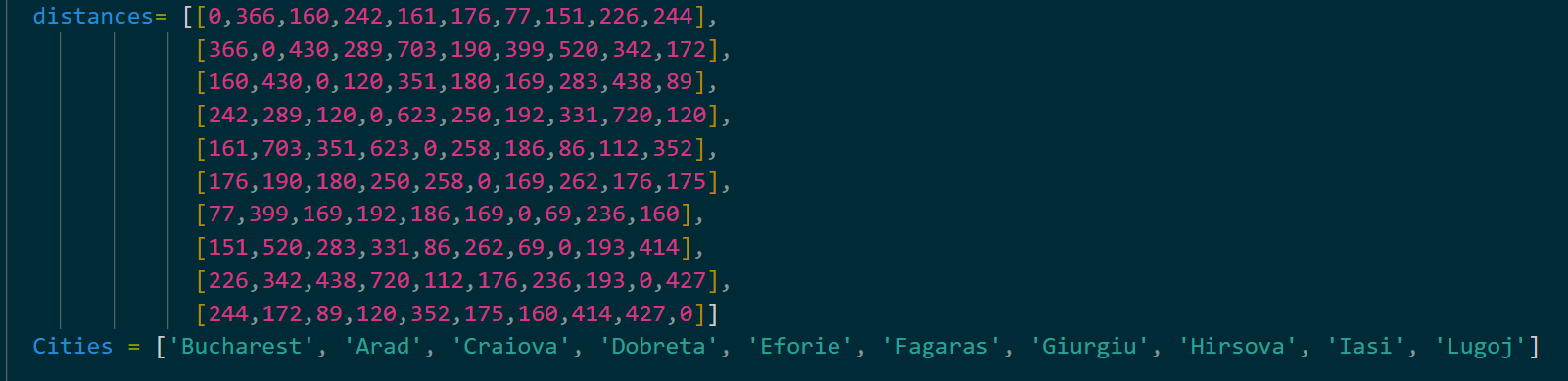
Tp. Hồ Chí Minh, ngày tháng năm

(ký và ghi họ tên)

**TÓM TẮT**

Bài toán người bán hàng:

Vấn đề nghiên cứu: Cho trước một danh sách các thành phố và khoảng cách giữa chúng, tìm chu trình ngắn nhất và dài nhất thăm mỗi thành phố đúng một lần.

Danh sách và tên các thành phố được thể hiện qua ma trận sau:

Hướng tiếp cận: sử dụng thuật toán di truyền để tìm đường đi dài nhất và ngắn nhất.

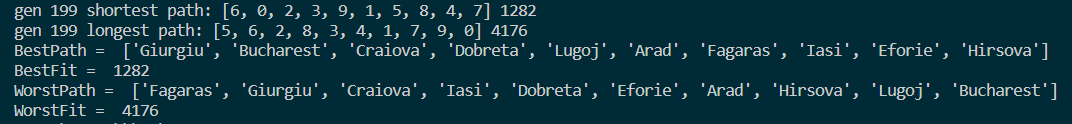
Cách giải quyết vấn đề:

Tạo ra một quần thể với 1000 tập hợp ngẫu nhiên các thành phố.

Tính toán độ tương tích ở ngưỡng thích hợp.

Lấy giá trị gần độ tương thích nhất và mang đi lai giống với tỉ lệ đột biến.

Kết quả đạt được:

Sau khoảng 200 thế hệ được kết quả sau:

Bài toán phát hiện va chạm:

Bài toán được tổ chức bời NFL yêu cầu lập trình một model thị giác máy tính có thể phát hiện va chạm của các cầu thủ bóng bầu dục và từ đó đánh giá sự an toàn của nón bảo hiểm và ngăn chặn chấn thương cho các cầu thủ.

Hướng tiếp cận:

Huấn luyện phát hiện nón bảo hiểm qua các hình ảnh 2D.

Huấn luyện va chạm từ các lớp 3D và từ dữ liệu 2D được huấn luyện.

**MỤC LỤC**

[**CHƯƠNG 1**](#_heading=h.jfwdvz9a69q0) **2**

[**Bài toán người bán hàng sử dụng thuật toán di truyền**](#_heading=h.4bq3fpbvm5pm) **2**

[1.1 Mô tả cấu trúc dữ liệu](#_heading=h.3vsaldc1j8lk) 2

[1.2 Sơ đồ giải thuật](#_heading=h.aoz5fsk2dvgm) 3

[1.3 Hiện thực](#_heading=h.jihtnhk0kprs) 11

[1.4 Kết quả và thảo luận](#_heading=h.xv6qwgsc205l) 15

[**CHƯƠNG 2**](#_heading=h.pk3o67rx6rza) **16**

[**NFL 1st and Future - Impact Detection**](#_heading=h.sgcota49pb16) **16**

[2.1 Giới thiệu về bài toán](#_heading=h.fyam2mes3610) 16

[2.2 Phân tích yêu cầu của bài toán](#_heading=h.1t3h5sf) 17

[2.2.1 Yêu cầu của bài toán](#_heading=h.c8di0upvhgn) 17

[2.2.2 Các phương pháp giải quyết bài toán](#_heading=h.j6h6clkp27s7) 17

[2.2.3 Phương pháp đề xuất giải quyết bài toán](#_heading=h.xi5sghnuiwb6) 17

[2.3 Phương pháp giải quyết bài toán](#_heading=h.j5lsdehusfaq) 18

[2.3.1 Mô hình tổng quát](#_heading=h.fjfglq81ycx9) 18

[2.3.2 Đặc trưng của mô hình đề xuất](#_heading=h.s9a79m8ctmv8) 18

[2.4 Thực nghiệm](#_heading=h.u3ewjiq7y1wp) 19

[2.4.1 Dữ liệu](#_heading=h.u7gm09ojpzx9) 19

[2.4.2 Xử lý dữ liệu](#_heading=h.e71u23jwa31r) 21

[2.4.3 Công nghệ sử dụng](#_heading=h.2vn4yevmnmt6) 32

[2.4.4 Cách đánh giá](#_heading=h.eupvu5karujt) 33

[2.5 Kết quả đạt được](#_heading=h.ror4bgx84pai) 33

[2.6 Kết luận](#_heading=h.2lm9yfa7u877) 34

[**TÀI LIỆU THAM KHẢO**](#_heading=h.7i5n9176rlvd) **35**

[**PHỤ LỤC**](#_heading=h.5wsy9r4xvsr5) **36**

[TỰ ĐÁNH GIÁ](#_heading=h.ua9ajavoqur9) 37

6

# CHƯƠNG 1

# Bài toán người bán hàng sử dụng thuật toán di truyền

## 1.1 Mô tả cấu trúc dữ liệu

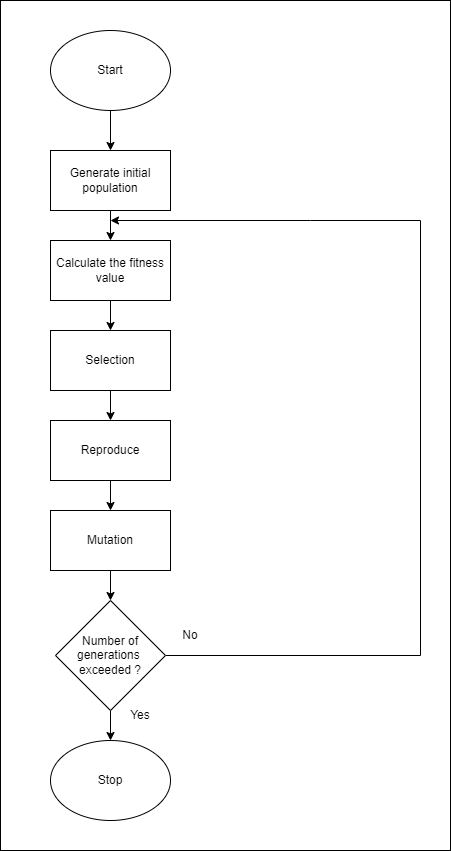
Thuật toán bắt đầu bằng cách khởi tạo một quần thể các tập hợp. Mỗi tập hợp là một tuple ngẫu nhiên gồm các thành phố sử dụng các giá trị số đại diện cho từng thành phố. Sau đó, nó tính toán đường đi từng của từng tập hợp của quần thể đó thông qua một **calculate fitness function(còn gọi là hàm thích nghi)**. Nó chọn các tập hợp có chỉ số **fitness** nhỏ nhất (lớn nhất) của quần thể để tái sản xuất (reproduce) bằng phương pháp được xác định trong **reproduce function (còn gọi là lai giống)**, bên trong **reproduce function** sẽ có phương pháp **mutation function(còn gọi là đột biến)** dựa trên tỷ lệ. Sau đó lặp lại việc đánh giá, tái sản xuất và đột biến cho đến khi đạt được số **generations (thế hệ)** mong muốn. Khi kết thúc, thuật toán đưa ra các thành phố có đường đi ngắn nhất (dài nhất). Chúng ta sẽ đi sâu hơn vào **calculate fitness function, reproduce function,** **reproduce function mutation function và generations (thế hệ)**

**calculate fitness function:** sẽ tính tổng các đường đi từ thành phố này qua thành phố khách bằng cách duyệt qua từng phần tử trong một tập hợp các thành phố.

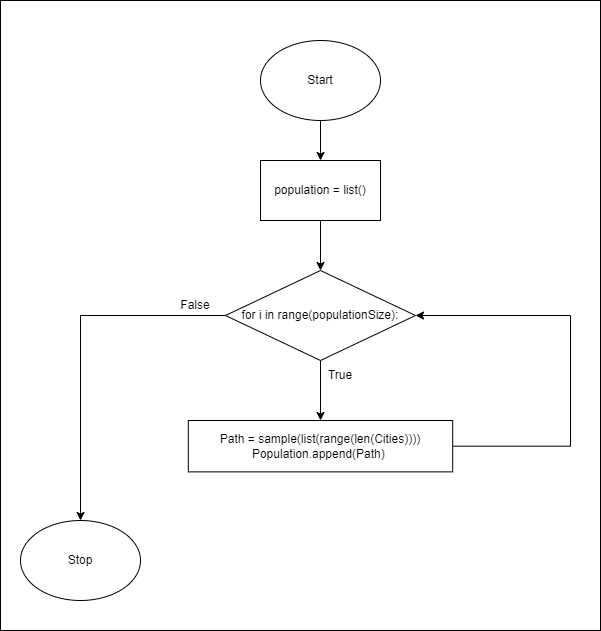
**reproduce function:** sẽ tái sản xuất từ 2 tập hợp ngẫu nhiên được chọn trong một quần thể cha các tập hợp tốt nhất dựa trên tổng các đường đi nhỏ nhất. Sau đó sẽ dựa trên tỷ lệ đột biến để đột biến (**mutation function)** tập hợp vừa được tái sản xuất bằng cách đổi chỗ 2 thành phố ngẫu nhiên bất kỳ.

**generations (thế hệ):** là quần thể con được tạo ra từ quần thể cha thông qua các bước **tái sản xuất(reproduce)** và **đột biến(mutation).**

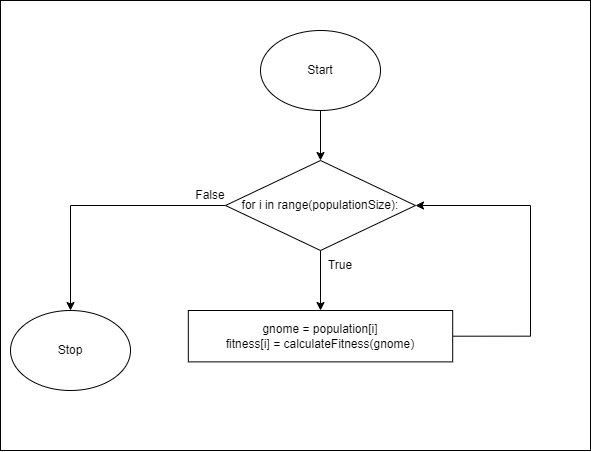
## **1.2 Sơ đồ giải thuật**



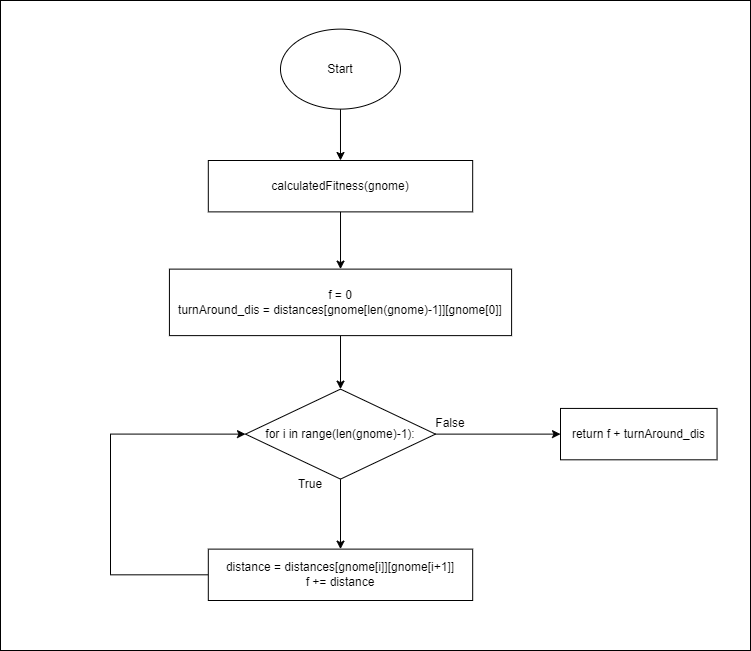
Hình 1.1: Sơ đồ khối giải thuật tổng quát



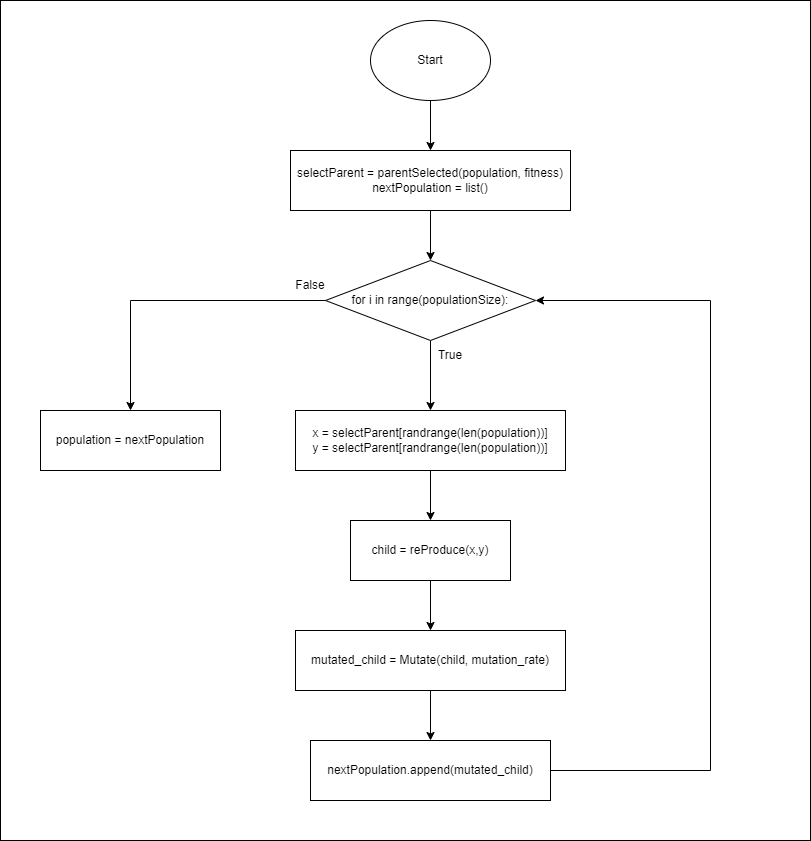
Hình 1.2: Generate initial Population



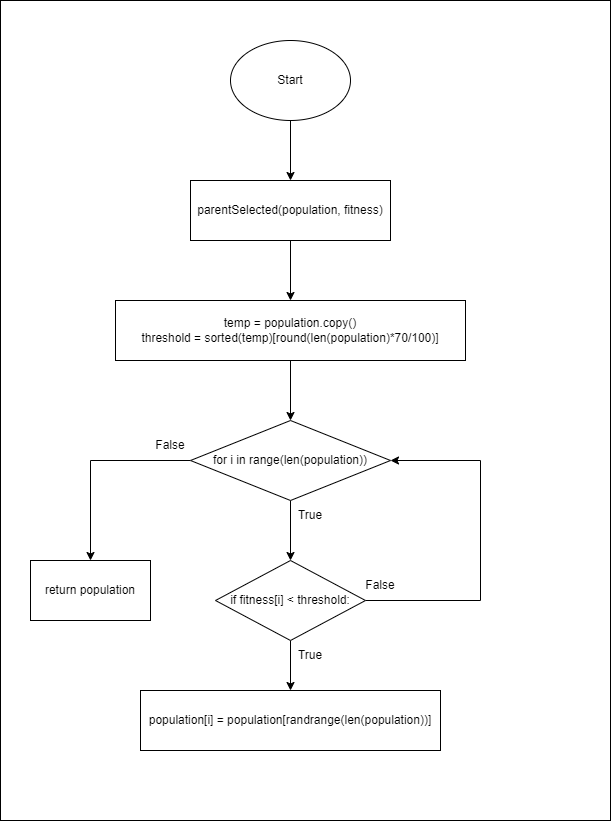
Hình 1.3: calculateFitness for each gnome in population



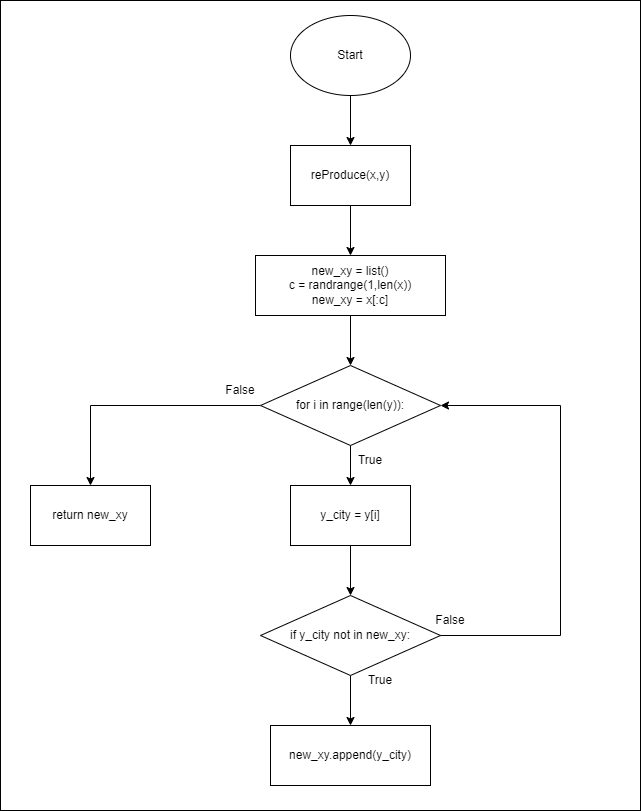
Hình 1.4: calculateFitness Function



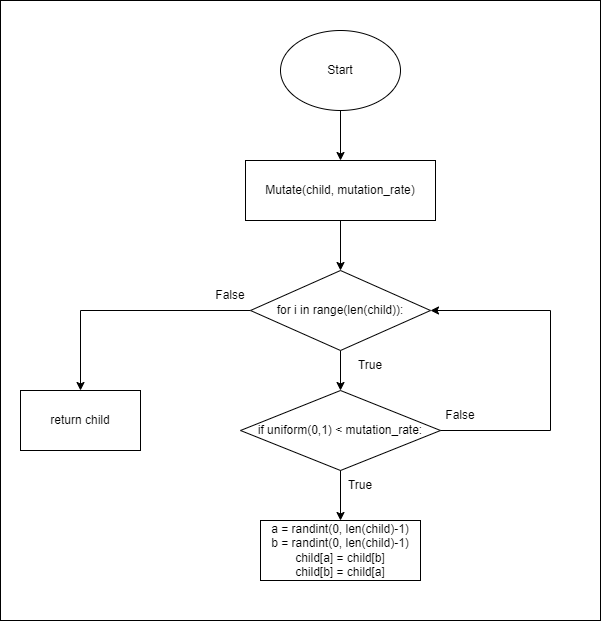
Hình 1.5: Selection, reProduc and mutation



Hình 1.6: Parent Selected function



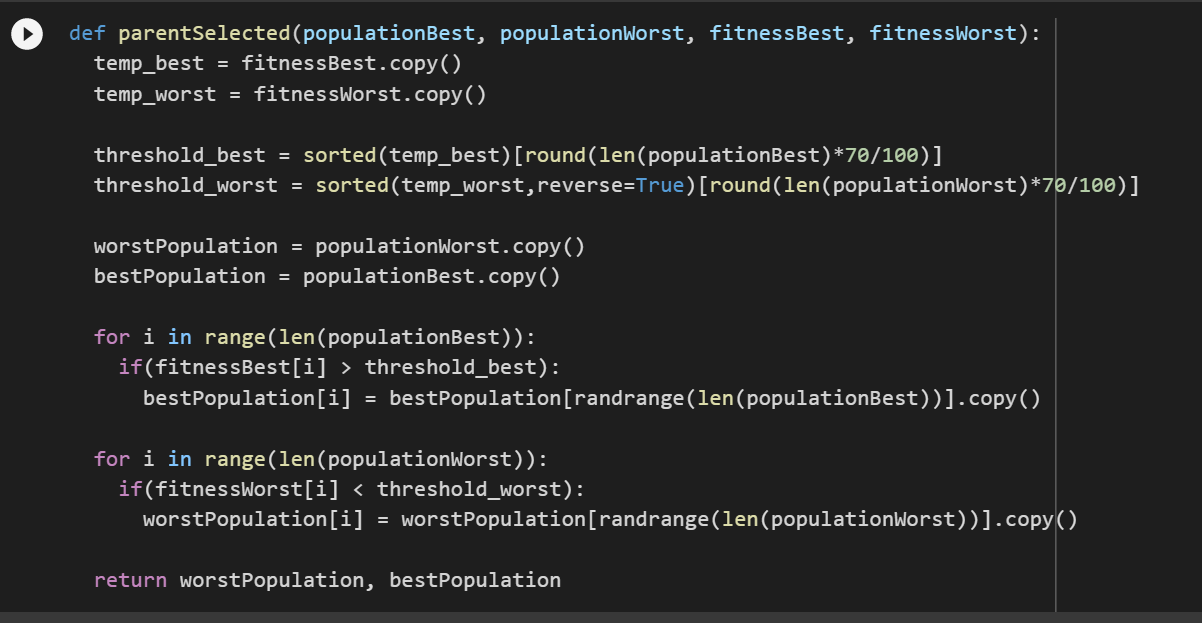
Hình 1.7: reProduce function



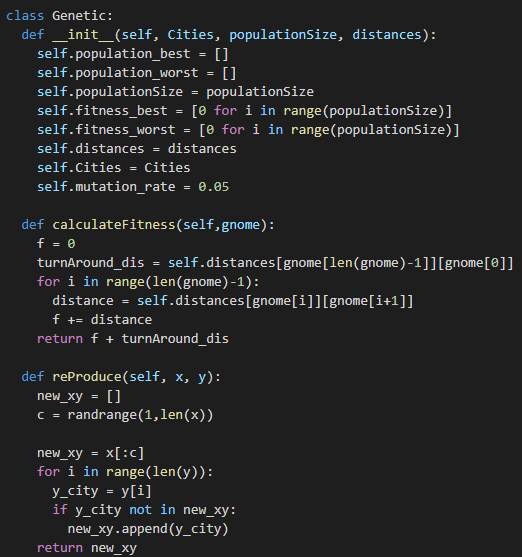
Hình 1.8: Mutate function

## 1.3 **Hiện thực**

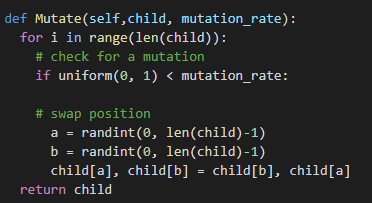
Trình bày code hiện thực của bài toán:



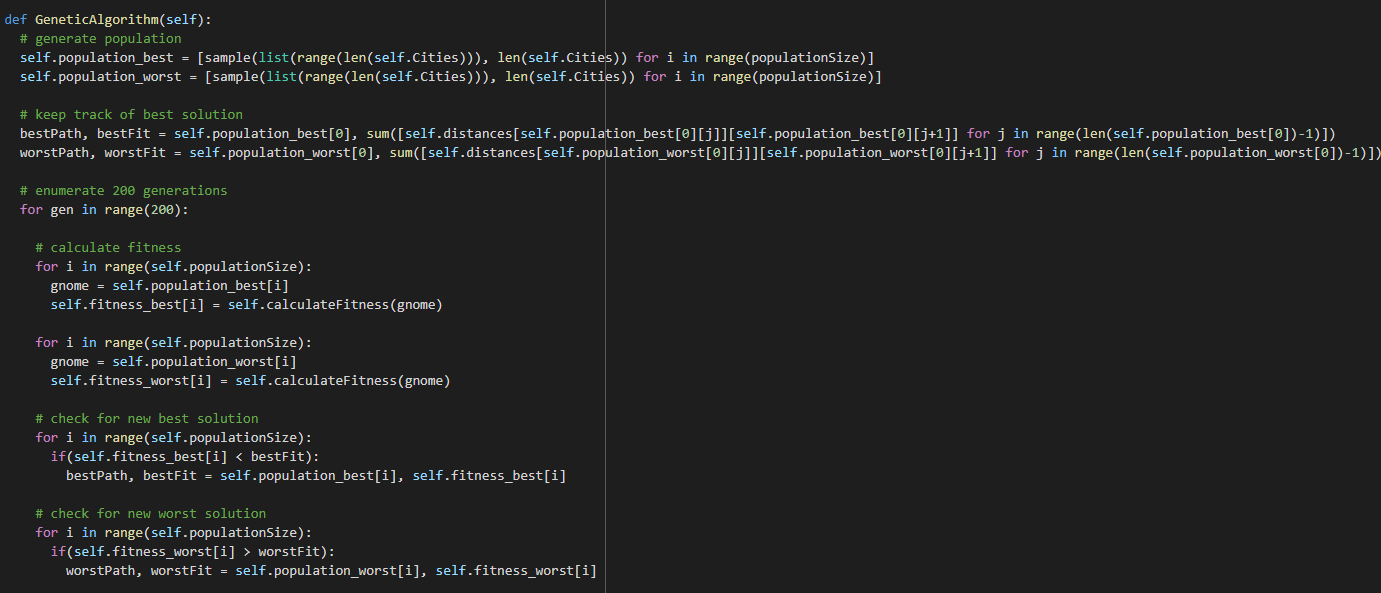
Hình : parent selection function



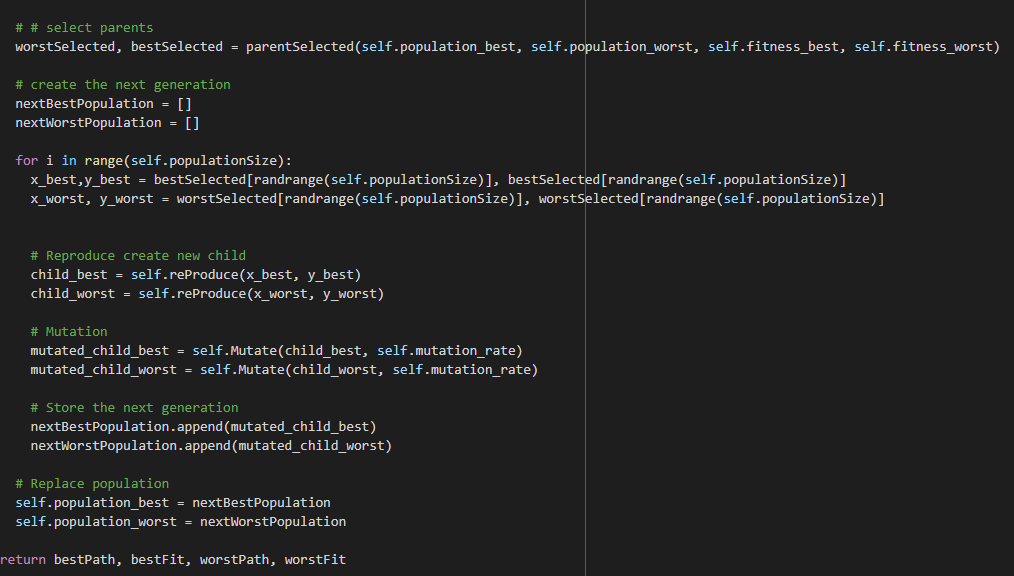
Hình 1.1: Code bao gồm hàm khởi tạo population, calculate fitness function và reproduce function.



Hình 1.2 bao gồm quá trình đột biến.



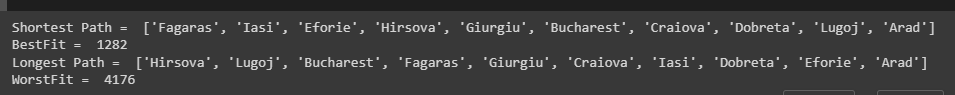
Hình 1.3: GeneticAlgorithm part 1



Hình 1.4: GeneticAlgorithm part 2

## 1.4 Kết quả và thảo luận

1. Trình bày kết quả đạt được



2. Trình bày các đánh giá giải thuật

Thuật toán phía trên chúng ta là O(g(nm + nm + n)) với g là số thế hệ, n là kích thước quần thể và m là kích thước của cá thể

3. Thảo luận về kết quả và giải thuật đã hiện thực:

Chúng ta có thể thấy được là thế hệ đầu tiên luôn có độ thích nghi thấp và không tốt nhưng càng về sau khi qua các đời thế hệ con thì độ thích nghi càng tốt và chính xác. Điều đó suy ra là bài toán di truyền qua càng nhiều thế hệ kết quả càng chính xác.

Bài toán tối ưu là bài toán tìm kiếm giải pháp (tốt nhất) trong không gian (vô cùng lớn) các giải pháp. Khi không gian tìm kiếm nhỏ, các phương pháp cổ điển cũng đủ thích hợp; nhưng khi không gian lớn hơn thì cần phải dùng đến những kĩ thuật đặc biệt, Genetic Algorithm là một trong những kĩ thuật đó. Genetic Algorithm là hệ thống đầu tiên dựa trên mô phỏng sự sống. Thông qua chọn lọc, lai ghép, đột biến, Genetic Algorithm sẽ hội tụ qua nhiều thế hệ theo hướng tối ưu toàn cục. Genetic Algorithm hi vọng hướng tới một giải pháp tối ưu hơn bằng cách kết hợp thông tin tốt ẩn trong tập hợp các giải pháp, để tạo ra giải pháp mới với những thông tin tốt thừa hưởng từ cả cha và mẹ. Khả năng của thuật toán vẫn đang còn được khám phá và khai thác, một số lượng lớn những nghiên cứu thành công về lí thuyết cũng như ứng dụng trong nhiều lĩnh vực thế giới thực đã chứng minh rằng Genetic Algorithm là một kĩ thuật tối ưu mạnh mẽ và thiết thực.

# CHƯƠNG 2

# NFL 1st and Future - Impact Detection

## 2.1 Giới thiệu về bài toán

Bài toán này tên là NFL 1st and Future - Impact Detection. Giải bóng bầu dục quốc gia(NFL) đã phối hợp với dịch vụ điện toán đám mây Amazon(AWS) để phát triển “Digital Athlete”, một giả lập các cầu thủ NFL sử dụng mode để tạo ra các tình cảnh để dự đoán và ngăn cản chấn thương của cầu thủ. NFL có nhu cầu thiết yếu là một thị giác máy tính để phát hiện các va chạm của nón bảo hiểm trong “Digital Athlete” platform.

Bài toán này có ý nghĩa gì?

Dùng để dự đoán sự va chạm của các cầu thủ từ đó cải thiện sự an toàn cho các vận động viên. Từ dự đoán này NFL sẽ thay đổi luật và thiết kế của nón bảo hiểm để bảo vệ các vận động viên hiệu quả hơn.

## 2.2 Phân tích yêu cầu của bài toán

### 2.2.1 Yêu cầu của bài toán

Phát triển một mô hình thị giác máy tính tự động phát hiện các va chạm nón bảo hiểm ở trên sân. Bắt đầu với 1000 va chạm từ hàng ngàn hình ảnh các trận đấu và các video được đặt quay từ góc sidelines và góc end zones, và theo dõi chuyển động của cầu thủ.

### 2**.2.2 Các phương pháp giải quyết bài toán**

Phương pháp tiếp cận bài toán đều tương tự nhau đầu tiên là sẽ nhận diện các nón bảo hiểm của vận động viên sau đó sẽ theo dõi và phân loại trường hợp va chạm hoặc không va chạm của nón bảo hiểm giữa các vận động viên.

Có 2 phương pháp được sử dụng để nhận diện nón bảo hiểm 2D:

* YoloV5-I (YOLO (You Only Look Once) dùng để nhận dạng đối tượng được thiết kế để nhận diện các vật thể real-time.)
* EfficientDet là một loại mô hình phát hiện đối tượng, sử dụng một số tinh chỉnh tối ưu hóa và xương sống, chẳng hạn như sử dụng BiFPN và phương pháp chia tỷ lệ hỗn hợp giúp chia tỷ lệ đồng đều độ phân giải, độ sâu và chiều rộng cho tất cả các xương sống, mạng tính năng và dự đoán hộp/lớp mạng đồng thời.

Có 2 phương pháp được sử dụng để nhận diện va chạm nón bảo hiểm 3D:

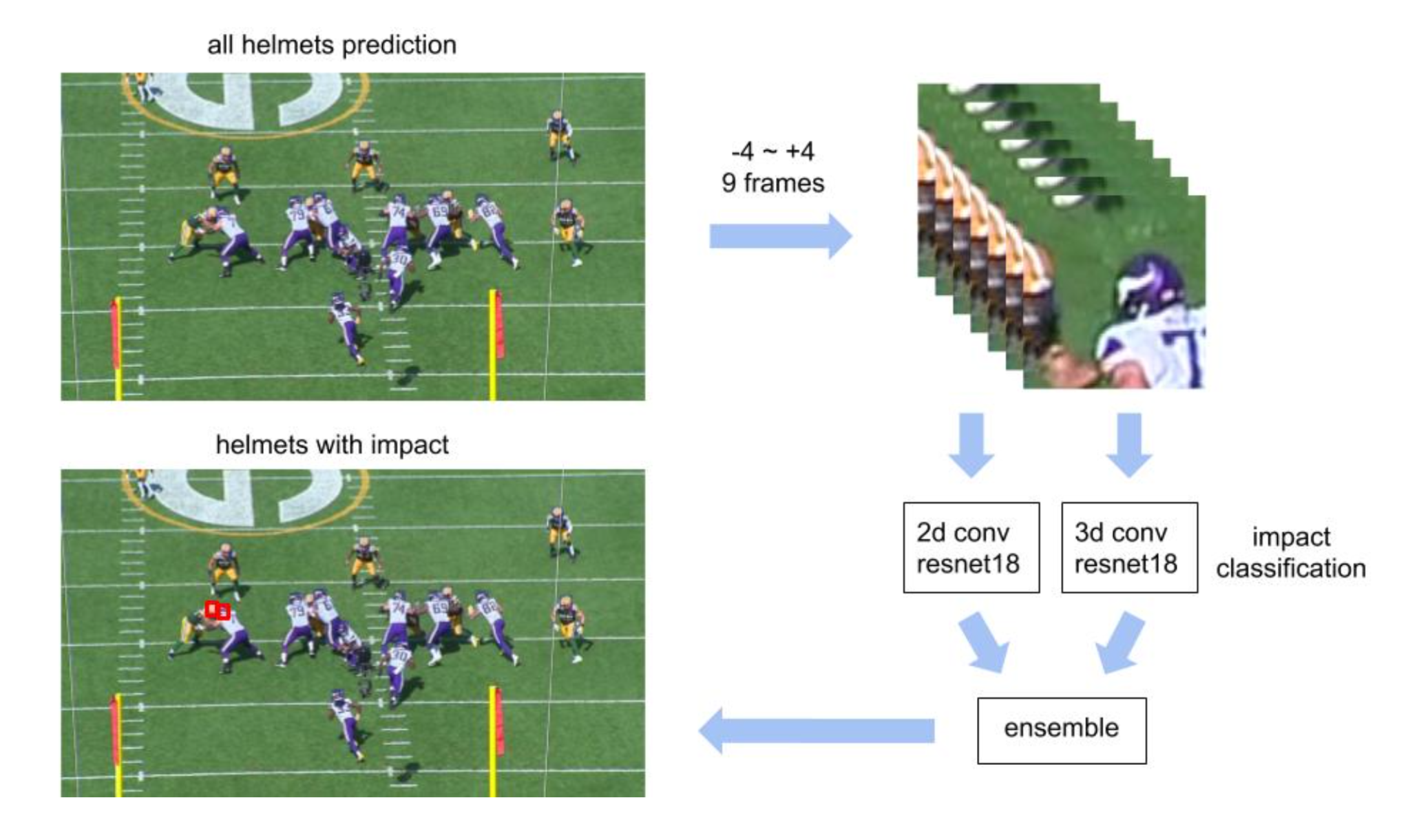
* Temporal Shift Moduel.
* 3D CNN model.
* EfficientDet + resnet18

### 2**.2.3 Phương pháp đề xuất giải quyết bài toán**

Sử dụng object detection eficientDet va resnet 18

## 2.3 Phương pháp giải quyết bài toán

### 2.3.1 Mô hình tổng quát

****

### 2.3.2 Đặc trưng của mô hình đề xuất

(Giai đoạn-1) Phát hiện tất cả mũ bảo hiểm với EfficientDet-d5

(Giai đoạn 2) Phân loại tất cả mũ bảo hiểm để tác động hoặc không tác động với resnet18

Mục đích dùng EfficientDet-d5 bởi vì chúng ta có thể cắt chúng thành các hình ảnh nhỏ. Sử dụng mô hình phân loại ảnh nhị phân để phân loại cây mũ bảo hiểm qua 9 khung hình.

huấn luyện resnet18 lên 2 lớp(va chạm hay không) và sử dụng nó để xác định nón bảo hiểm đã va chạm nhau.

Để huấn luyện đầu vào cho resnet18, cắt hình ảnh xung quanh nón bảo hiển và chuẩn bị 9 hình ảnh từ 4 frames trước và sau.

Hình ảnh được cắt gấp 3 lần phía dài nhất của hình chữ nhật quanh nón bảo hiểm (max(w,h) \* 3).

Hình ảnh được cắt được chỉnh lại thành hình vuông và chỉnh thành 112\*112.

Ở trạng thái, hình ảnh được huấn luyện được lấy ra từ video mỗi 10 frame và chủ lấy khung hinh xảy ra va chạm.

Khi tạo ra dữ liệu huấn luyện, cách đánh giá để xác định va chạm xảy ra hay không (impact = 1, confidence > 1, visibility > 0)

## 2.4 Thực nghiệm

### 2.4.1 Dữ liệu

Dữ liệu được cung cấp sẵn bởi NFL: https://www.kaggle.com/competitions/nfl-impact-detection/data

Dữ liệu gồm có:

**train\_labels.csv** Theo dõi nón bảo hiểm và tên va chạm cho training set.

* gameKey: ID code cho trận đấu.
* playID: the ID code for the play.
* view: hướng camera
* video: tên video.
* frame: số khung hình.
* label:số của người chơi.
* [left/width/top/height]: cụ thể của khung hình.
* impact: impact = 1 nghĩa là có xảy ra va chạm.
* impactType: thể loại va chạm bảo hiểm, vai, người, đất.
* confidence: 1 = Có thể, 2 = Chắc chắn, 3 = Chắc chắn và hiển nhiên.
* visibility: 0 = không thể nhìn thấy, 1 = Khó thấy, 2 = Thấy, 3 = Thấy rõ ràng

**sample\_submission.csv** A valid sample submission file.

* gameKey: the ID code của game.
* playID: the ID code lúc chơi bóng.
* view: hướng camera.
* video:tên video.
* frame: số frame.
* [left/width/top/height]:cụ thể cho dự đoán của hộp.

**images** hình ảnh dataset.

**image\_labels.csv** chứa các hộp giới hạn tới của các hình ảnh.

* image: tên hình ảnh.
* label: tên.
* [left/width/top/height]: .

**[train/test]\_player\_tracking.csv** Mỗi người chơi có cảm biến để chúng ta dò theo.

* gameKey: the ID code cho trận đấu.
* playID: the ID code cho pha xử lí.
* player: ID code của người chơi.
* time: timestamp tại 10 Hz.
* x: vị trí của người chơi trục ngang của sân đấu.
* y: vị trí của người chơi trục dọc của sân đấu.
* s: tốc độ y/s.
* a:gia tốc y/s^2.
* dis: khoảng cách di chuyển.
* o: góc độ của người chơi.
* dir: gốc độ di chuyển.
* event: sự kiện của trận đấu.

### 2.4.2 Xử lý dữ liệu

Có phải tiền xử lý dữ liệu không? Tại sao

Phải xử lí dữ liệu vì tất cả đều là bản thô, gồm hình và video của các trận đấu. Mục đích xử lí dữ liệu đầu tiên là xác định nón bảo hiểm của người chơi và sau đó xác định va chạm của nón bảo hiểm.

Xử lý dữ liệu như thế nào?

Để huấn luyện, sử dụng các khung hình xảy ra va chạm và các khung hình không xảy ra va chạm. Hình ảnh được huấn luyện lấy được lấy ra từ video mỗi 10 khung hình và +-5 các khung hình từ khung hình xảy ra va chạm.

Các hình ảnh được phóng đại 1280 \* 1280 và được xoay dọc để tăng thêm bởi vì

tổn thất hội tự nhanh hơn.

Kết luận được thực hiện lên 2 hình ảnh, hình ảnh gốc và hình ảnh xoảy dọc. Cuối cùng kết quả được kết hợp sử dụng wbf.

Huấn luyện resnet18 lên 2 lớp(va chạm hay không) và sử dụng nó để xác định nón bảo hiểm đã va chạm nhau.

Để huấn luyện đầu vào cho resnet18, cắt hình ảnh xung quanh nón bảo hiển và chuẩn bị 9 hình ảnh từ 4 frames trước và sau.

Hình ảnh được cắt gấp 3 lần phía dài nhất của hình chữ nhật quanh nón bảo hiểm (max(w,h) \* 3).

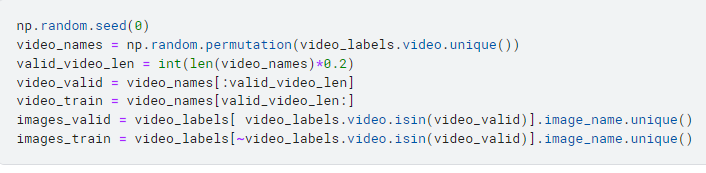
Hình ảnh được cắt được chỉnh lại thành hình vuông và chỉnh thành 112\*112.

Ở trạng thái, hình ảnh được huấn luyện được lấy ra từ video mỗi 10 frame và chủ lấy khung hinh xảy ra va chạm.

Khi tạo ra dữ liệu huấn luyện, cách đánh giá để xác định va chạm xảy ra hay không (impact = 1, confidence > 1, visibility > 0)







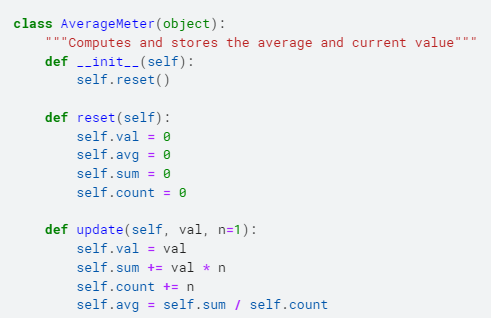
















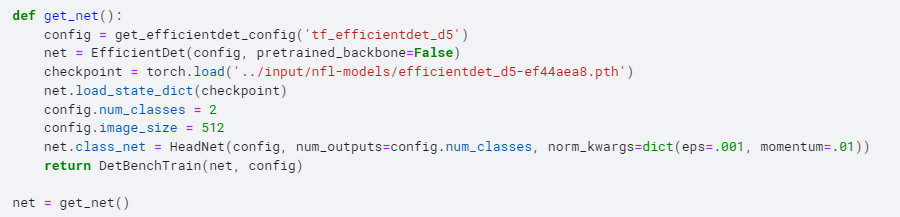








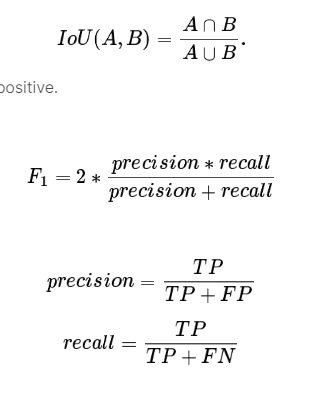




### 2.4.3 Công nghệ sử dụng

Công nghệ sử dụng chủ yếu là notebook của kaggle với ngôn ngữ chủ yếu là python.

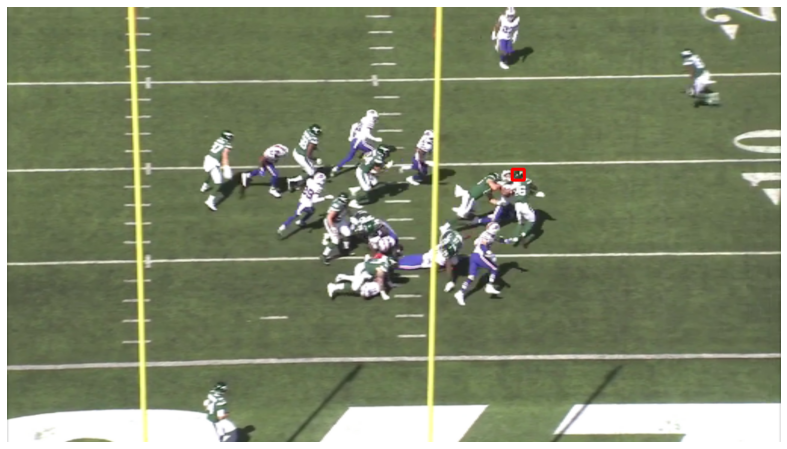
### **2.4.4 Cách đánh giá**

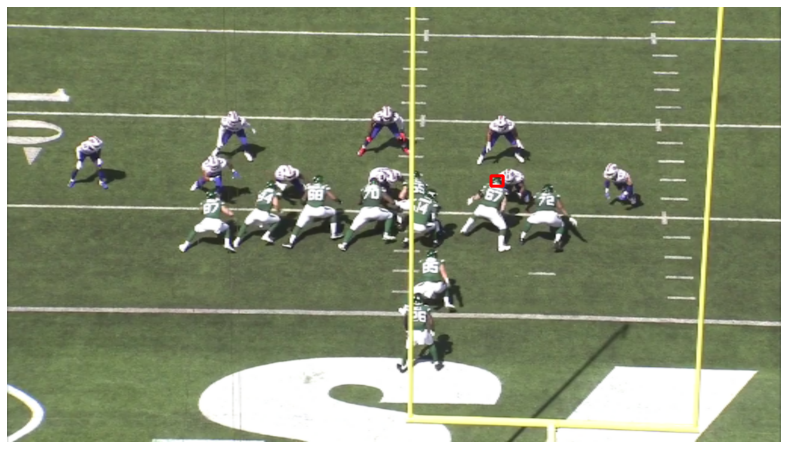


cách đánh giá để xác định va chạm xảy ra hay không (impact = 1, confidence > 1, visibility > 0)

## 2.5 Kết quả đạt được

****

****

****

## 2.6 Kết luận

Kết quả đạt được:

Mô hình phát hiện được chính xác trên khung hình nón bảo hiểm, va chạm nón bảo hiểm của các người chơi sau khi được huấn luyện dựa trên các dataset được cung cấp.

# TÀI LIỆU THAM KHẢO

[**https://www.kaggle.com/code/khanhtrn/2class-object-detection-inference/edit**](https://www.kaggle.com/code/khanhtrn/2class-object-detection-inference/edit)

[**https://www.kaggle.com/code/its7171/2class-object-detection-training/data**](https://www.kaggle.com/code/its7171/2class-object-detection-training/data)

[**https://www.kaggle.com/competitions/nfl-impact-detection/discussion/209012**](https://www.kaggle.com/competitions/nfl-impact-detection/discussion/209012)

[**https://www.kaggle.com/competitions/nfl-impact-detection/discussion/209012**](https://www.kaggle.com/competitions/nfl-impact-detection/discussion/209012)

[**https://www.youtube.com/watch?v=5jyBcxrtE1Y**](https://www.youtube.com/watch?v=5jyBcxrtE1Y)

**https://www.kaggle.com/code/ultralytics/yolov5**

# PHỤ LỤC

## TỰ ĐÁNH GIÁ

(Với nhóm có 2 thành viên)

| Câu | Nội dung | Điểm chuẩn | Tự chấm | Ghi chú |
| --- | --- | --- | --- | --- |
| 1  (4.0) | **1.1 Mô tả cấu trúc dữ liệu** | 0.5 | 0.5 |  |
| **1.2 Sơ đồ giải thuật** | 1 | 1 |  |
| **1.3 Hiện thực** | 2 | 2 |  |
| **1.4 Kết quả và thảo luận** | 0.5 | 0.5 |  |
| 2  (4.5) | **2.1 Giới thiệu về bài toán** | 0.25 | 0.25 |  |
| **2.2 Phân tích yêu cầu của bài toán** | 1 | 0.75 |  |
| **2.3 Phương pháp giải quyết bài toán** | 1.25 | 0.75 |  |
| **2.4 Thực nghiệm** | 1 | 0.5 |  |
| **2.5 Kết quả đạt được** | 0.75 | 0.25 |  |
| **2.6 Kết luận** | 0.25 | 0 |  |
| 3 | **Điểm nhóm** | 0.5đ | 0.5 |  |
| 4 | **Báo cáo (**chú ý các chú ý 2,3,4,6 ở trang trước, nếu sai sẽ bị trừ điểm nặng**)** | 1đ | 1đ |  |
| **Tổng điểm** | | | 8 |  |