

NHÓM 5

THÀNH VIÊN:

Nguyễn Quốc Huy Hoàng – 20520051

Lê Nguyễn Khánh Nam – 20520073

CHỦ ĐỀ: COMPUTATIONAL THINKING

Câu hỏi:	Câu 1: <i>Input/output của mỗi kỹ thuật trong computational thinking?</i>		Câu 2: <i>Những câu hỏi nào thường được đặt ra để thực hiện các kỹ thuật computational thinking?</i>	Bài toán: <i>Xây dựng mô hình cảnh báo khoảng cách giữa mọi người trong 1 tòa nhà dựa trên hình ảnh từ các camera cố định.</i>
Các kỹ thuật	Input	Output	Câu hỏi thường đặt ra	Áp dụng bài toán
Abstraction Tóm tắt	<ul style="list-style-type: none"> Các dữ liệu của vấn đề, bài toán, trong đó có nhiều thông tin nhiễu, nhiều thành phần không quan trọng. 	<ul style="list-style-type: none"> Thông tin đã được cô đọng lại. Có được các điểm quan trọng như là các điểm quan trọng cần chú ý, yêu cầu đầu ra của vấn đề. 	<ul style="list-style-type: none"> Tham số đầu vào là gì? Giới hạn của các tham số của input là bao nhiêu? Yêu cầu của đầu ra là gì? Vấn đề trọng tâm mà ta phải quan tâm là gì? 	<ul style="list-style-type: none"> Input: Dữ liệu từ camera (Video) Output: Đưa ra mô hình cảnh báo nếu có ít nhất 2 người có khoảng cách không đủ an toàn (khoảng cách không đủ an toàn có thể là 2m chẳng hạn).
Pattern recognition Nhận diện mẫu	<ul style="list-style-type: none"> Bài toán, vấn đề chưa có cách giải quyết hoặc cách giải quyết quá chưa tốt. 	<ul style="list-style-type: none"> Đưa ra được các đặc điểm nổi bật, tính chất quan trọng, quy luật của bài toán. 	<ul style="list-style-type: none"> Vấn đề này có những điểm gì đặc biệt? Nó sẽ thuộc kiểu bài toán như thế nào? 	<ul style="list-style-type: none"> Bài toán này sử dụng input từ video nên liên quan đến Computer Vision Camera thường quay từ trên xuống, có thể coi hình ảnh camera quay được là hình ảnh trên mặt phẳng 2D (Oxy), mỗi người giống như 1 điểm trên đó. Ta tính được khoảng cách dựa vào khoảng cách giữa các pixel của ảnh.

	Input	Output	Câu hỏi thường đặt ra	Áp dụng bài toán
Decomposition Phân rã	<ul style="list-style-type: none"> - Một vấn đề, bài toán phức tạp, không thể giải quyết hết toàn bộ. 	<ul style="list-style-type: none"> - Nhiều vấn đề, bài toán con phân chia từ bài toán gốc, có thể giải quyết, quản lý được. 	<ul style="list-style-type: none"> - Nó có thể được giải quyết bằng những cách nào? - Nó có thể phân ra được các bài toán nhỏ nào để giúp cho việc tính toán dễ dàng hơn không? 	<ul style="list-style-type: none"> - Dữ liệu đầu vào là video nên ta phải tách video thành các frame, độ dài 1s thì lấy cỡ 5 frames chẳng hạn (tiền xử lý). - Để xác định được khoảng cách thì ta phải có các đối tượng (ở đây là người), tiến hành detect những người có trong frame và gắn nhãn và tracking => Có thể dùng Object detection và object tracking.
Algorithm design Thiết kế thuật toán	<ul style="list-style-type: none"> - Một bài toán chưa có hướng giải quyết cụ thể (Có thể là bài toán gồm nhiều bài toán con rời rạc nhau, hoặc là bài toán đã biết được các tính chất đặc biệt nhưng chưa có trình tự thực hiện rõ ràng). Hoặc vấn đề nào đó đã có hướng giải pháp nhưng chưa được tốt hoặc không khả thi. 	<ul style="list-style-type: none"> - Lưu đồ, mô hình thuật toán, sao cho có thể giải quyết được bài toán một cách phù hợp nhất (Có thể là tốt về độ chính xác, tốt về mặt thời gian, cách cài đặt) 	<ul style="list-style-type: none"> - Dựa vào những đặc điểm mà mình đã có thì những thuật toán nào có thể giải quyết được nó? - Liệu thuật toán này có thực sự phù hợp với vấn đề này hay chưa? 	<ul style="list-style-type: none"> - Camera có thể đặt ở nhiều góc độ khác nhau ở các video khác nhau, nên mỗi pixel ở điểm xa camera ta tính ra khoảng cách lớn hơn khi quy đổi. (khi kết quả testing chưa tốt thì điều chỉnh cách quy đổi cho phù hợp) - Khi ta đã nhận dạng, đánh dấu vị trí rồi thì ta có thể tính được khoảng cách giữa 2 người bất kì. Do số người trong một tòa nhà không lớn nên có thể dùng Brute force duyệt qua hết tất cả các cặp. - Với 2 người trong nhiều frame liên tục mà vi phạm khoảng cách thì cảnh báo (nghĩa là trong thời gian đủ lâu, có thể điều chỉnh, do mình tự quy định)

	Input	Output	Câu hỏi thường đặt ra	Áp dụng bài toán
Testing Kiểm tra	<ul style="list-style-type: none"> - Output từ bài toán và Output từ thuật toán có được sau khi nạp cùng một tập Input . 	<ul style="list-style-type: none"> - Tùy vào problem mà có thể là Accuracy hay là True/ False 	<ul style="list-style-type: none"> - Thuật toán mình áp dụng có độ chính xác như thế nào? - Liệu có thể đáp ứng được yêu cầu mà đề bài đưa ra hay không? 	<ul style="list-style-type: none"> - Ta không có đáp án của dữ liệu vào, nên ta phải tự tính hoặc tạo ra dữ liệu mới để test. Trường hợp này thì ta tự quay nhiều video ở nhiều trường hợp khác nhau, sau đó lấy video đó làm input để so kết quả. - Kiểm tra thuật toán mất bao lâu để cho ra out kể từ khi ta cho input vào. - Nếu kết quả test cho ra không được tốt thì ta quay lại bước Algorithm design để bổ xung cũng như là điều chỉnh lại dữ liệu và điều kiện cho phù hợp.
Evaluation Đánh giá, ĐPT	<ul style="list-style-type: none"> - Tất cả các thuật toán của các bài toán con sau khi được phân rã (Decomposition) từ bài toán gốc và thuật toán chính. 	<ul style="list-style-type: none"> - Tổng độ phức tạp của tất cả thuật toán (Độ phức tạp trường hợp tệ nhất, trường hợp trung bình). Cũng như đánh giá được tính khả dụng, độ ổn định. 	<ul style="list-style-type: none"> - Độ phức tạp của thuật toán sẽ là bao nhiêu? - Độ phức tạp trung bình, xấu nhất? - Tính khả thi thế nào? 	<ul style="list-style-type: none"> - Để khi đưa ra mô hình hoạt động được trơn tru và chính xác thì ta cần tính những trường xấu nhất của thể xảy ra. Ở đây dùng thuật toán Brute force để kiểm tra khoảng cách, nên khi số người tăng lên thì độ phức tạp cũng tăng lên rất lớn. Ta tìm nhiều video có số lượng người lớn, không cần biết kết quả đầu ra thế nào, mục đích chính là để kiểm tra xem nó hoạt động có trơn tru không. Nếu không ổn thì quay lại bước Algorithm design để cải tiến.