



## XÂY DỰNG TRÒ CHƠI TIC-TAC-TOE VỚI NGƯỜI ĐÁU VỚI MÁY TRONG PHIÊN BẢN THỰC TẾ TĂNG CƯỜNG

Lê Minh Hưng và Phạm Nguyên Khang

Khoa Công nghệ Thông tin và Truyền thông, Trường Đại học Cần Thơ

### Thông tin chung:

Ngày nhận bài: 15/09/2017

Ngày nhận bài sửa: 10/10/2017

Ngày duyệt đăng: 20/10/2017

### Title:

Building tic-tac-toe game with human and computer players in augmented reality version

### Từ khóa:

Chơi cờ tự động, giải thuật cắt tỉa Alpha – Beta, kỹ thuật khung xương hóa, mô hình phân lớp phân tầng sử dụng giải thuật AdaBoost, phép biến đổi Hough, thực tế tăng cường, trò chơi tic-tac-toe

### Keywords:

Alpha – Beta pruning algorithm, Augmented Reality, auto-chess, cascade classifier using AdaBoost algorithm, skeletonization technique, tic-tac-toe game, Hough transform

### ABSTRACT

Augmented reality is a technology that combines digital information and real world in real time, input data is recorded through the camera of devices such as phone, laptop, etc. The information is usually enhanced on 3D objects, video, audio, etc. This paper presented the construction of tic-tac-toe game with human and computer players applied augmented reality technology. Image processing techniques and Hough transformation were used to detect 4 straight lines of the checkerboard, from which 9 squares were extracted. To identify the 'X' that the player hits, the cascade classifiers were used, each cascade classifier is a Adaboost algorithm. The experiments showed that the accuracy of 'X' mark recognition is above 98%. As for the problem of auto-chess, the Alpha – Beta pruning algorithm was applied.

### TÓM TẮT

Thực tế tăng cường (Augmented Reality - AR) là một công nghệ kết hợp giữa thông tin kỹ thuật số và thế giới thực trong thời gian thực, dữ liệu đầu vào được ghi nhận thông qua camera của các thiết bị như điện thoại, laptop, ... Các thông tin được tăng cường thường là đối tượng 3D, video, âm thanh, ...

Trong bài báo này, chúng tôi trình bày về việc xây dựng một chương trình chơi trò chơi tic-tac-toe với người đấu với máy ứng dụng công nghệ thực tế tăng cường. Sử dụng kỹ thuật xử lý ảnh và phép biến đổi Hough để phát hiện 4 đường của bàn cờ, từ đó trích xuất ra được 9 ô cờ. Đó sẽ là cơ sở để chúng tôi có thể phát hiện được dấu 'X' mà người chơi đi, chúng tôi sử dụng mô hình phân lớp phân tầng với mỗi tầng là bộ phân lớp AdaBoost, qua thực nghiệm, kết quả nhận dạng chính xác đạt hơn 98%. Để ra nước đi của máy một cách "thông minh", chúng tôi áp dụng giải thuật cắt tỉa Alpha-Beta.

Trích dẫn: Lê Minh Hưng và Phạm Nguyên Khang, 2017. Xây dựng trò chơi tic-tac-toe với người đấu với máy trong phiên bản thực tế tăng cường. Tạp chí Khoa học Trường Đại học Cần Thơ. Số chuyên đề: Công nghệ thông tin: 19-24.

## 1 GIỚI THIỆU

Ngày nay, trí tuệ nhân tạo đang là tâm điểm nghiên cứu của các nhà khoa học máy tính, bởi lợi ích mà trí tuệ nhân tạo mang lại, được ứng dụng rộng rãi trong nhiều lĩnh vực của cuộc sống thường ngày như thương mại, các ứng dụng trợ lý ảo dành

cho điện thoại thông minh cho đến các vấn đề mang tính học thuật cao như hệ chuyên gia, người máy học (robotics). Ngoài các lĩnh vực đã liệt kê, không thể không nhắc đến lĩnh vực giải trí mà ở đó với trí thông minh nhân tạo, máy tính có thể đấu với con người trong các trò chơi mang tính chiến lược.

Trong lĩnh vực giải trí, để tạo sức hút cho các trò chơi chiến thuật, bên cạnh ứng dụng trí tuệ nhân tạo được xem như “bộ não” của chương trình thì vấn đề then chốt là giao diện và phương thức giao tiếp với chương trình. Các chương trình trò chơi truyền thống không còn tạo nhiều sức hút đối với người chơi, mà thay vào đó, công nghệ thực tế tăng cường đang tạo nên một xu hướng mới cho các nhà phát triển trò chơi. Bởi luật chơi đơn giản và có tính tổng quát cao, dễ dàng tiếp cận nên chúng tôi đã lựa chọn trò tic-tac-toe để ứng dụng công nghệ thực tế tăng cường và chơi game tự động.

Đã có nhiều nghiên cứu về đề tài thực tế tăng cường áp dụng trí tuệ nhân tạo trong lĩnh vực trò chơi, tuy nhiên có khá ít đề tài xây dựng với trò tic-tac-toe. Trong báo cáo của (Maguire and Saltzman, 2013) chương trình nhận dạng khá tốt dấu ‘X’ của người dùng, tốc độ ra nước đi của máy nhanh, tuy nhiên do việc bàn cờ bị che khuất khi người chơi đi nước cờ nên khi ấy, máy tính không thể nhận dạng được bàn cờ và việc xử lý bị gián đoạn, từ đó ảnh hưởng đến việc hiển thị bàn cờ, làm gián đoạn việc chơi cờ của người chơi. Ở bài báo của (Klinker *et al.*, 1999), người chơi đi nước cờ của mình bằng vật thể có hình dạng và kích thước tương tự con cờ vậy, hệ thống có màn hình hiển thị thông báo đang là lượt đi của ai? Và thông báo kết quả khi kết thúc ván cờ. Tuy nhiên, có một bất tiện là khi người dùng hoàn thành nước đi của mình thì họ phải chạm tay vào phím ảo “Go” hoặc nói “Go” vào microphone được kết nối với hệ thống, do chương trình chưa thể tự động nhận biết khi nào người chơi đã hoàn thành nước đi của mình.

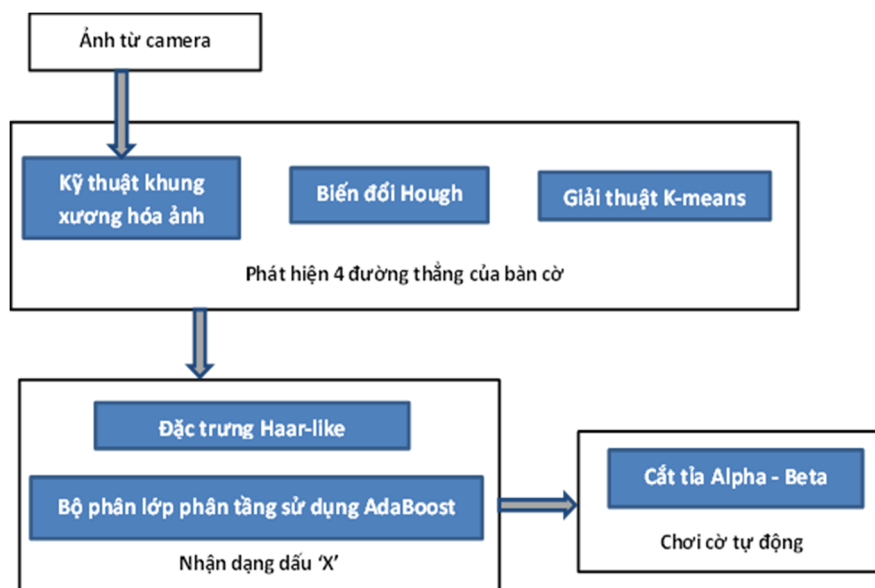
Trong bài báo này, chúng tôi sẽ trình bày khả năng nâng cao nhận dạng chính xác bàn cờ, cũng như dấu ‘X’ của người chơi đánh ra và máy tính có thể đưa ra các nước đi một cách “thông minh”.

Phần tiếp theo của bài báo được tổ chức như sau: phần 2 mô tả sơ lược về hệ thống, trình bày ngắn gọn về kỹ thuật khung xương hóa (skeletonization) đối tượng ảnh kết hợp với phép biến đổi Hough và giải thuật gom cụm K-means để phát hiện đường thẳng của bàn cờ; giới thiệu đặc trưng Haar-like cùng với mô hình phân lớp phân tầng sử dụng giải thuật AdaBoost để có thể nhận dạng được dấu ‘X’ của người chơi đánh ra; trình bày sơ lược về giải thuật cắt tia Alpha – Beta nhằm làm cho máy tính có thể đưa ra nước đi một cách “thông minh”; phần 3 trình bày về kết quả thực nghiệm trước khi kết luận và hướng phát triển được trình bày trong phần 4.

## 2 PHƯƠNG PHÁP NGHIÊN CỨU

### 2.1 Mô tả sơ lược về hệ thống

Hình ảnh bàn cờ được thu lại từ camera của laptop hoặc máy tính để bàn sẽ được phát hiện 4 đường thẳng tạo nên bàn cờ bằng cách áp dụng kỹ thuật khung xương hóa (skeletonization) ảnh để tìm khung xương của 4 đường thẳng, sau đó dùng phép biến đổi Hough để tìm ra 4 đường thẳng ấy. Ở đây có một vấn đề nổi lên là các đường thẳng được phát hiện qua phép biến đổi Hough không phải là 1-1 so với thực tế mà là 1 đường thẳng trong thực tế có thể được phát hiện thành nhiều hơn 1 trong phép biến đổi Hough, có đặc điểm là các đường thẳng ấy lân cận nhau, do vậy, chúng tôi đề xuất áp dụng giải thuật gom cụm K-means để giải quyết vấn đề này.



Hình 1: Mô tả tổng quan hệ thống

Sau khi đã phát hiện chính xác 4 đường thẳng tạo nên bàn cờ, đó sẽ là cơ sở để chúng tôi trích xuất ra 9 ô cờ, và cũng từ đây, chúng tôi cần nhận dạng dấu ‘X’ mà người chơi đã đi, chúng tôi sử dụng đặc trưng Haar-like và bộ phân lớp phân tầng mỗi tầng là 1 bộ phân lớp AdaBoost để có thể nhận dạng dấu ‘X’. Sau khi đã nhận dạng được dấu ‘X’ do người chơi đánh vào sẽ đến lượt của máy tính đưa ra nước đi của mình, để máy tính có thể đưa ra các nước đi “thông minh” thì chúng tôi áp dụng giải thuật cắt tỉa Alpha – Beta.

## 2.2 Phát hiện đường thẳng trong bàn cờ

Ở giai đoạn này, chúng tôi sẽ áp dụng các phép toán xử lý hình thái cơ bản trong xử lý ảnh như phép toán ăn mòn, phép toán giãn nở và phép toán mở để có thể khung xương hóa (skeletonization) ảnh nhị phân đầu vào. Khung xương (skeleton) (Kresch and Malah, 1998), (Magaros *et al.*, 1986) của đối tượng ảnh nhị phân là tập hợp các đường nét mỏng mà chúng chỉ giữ lại các thông tin quan trọng về hình dạng của đối tượng ban đầu. Khung xương hóa là quá trình mà ta thực hiện để tìm ra khung xương của đối tượng ảnh. Nó còn có tên gọi khác là phép biến đổi trục ở giữa (medial axis transform).

Sau khi đã tìm được khung xương, chúng tôi áp dụng phép biến đổi Hough để có thể phát hiện ra 4 đường thẳng tạo nên bàn cờ. Phép biến đổi Hough (Hough, 1962) dùng để phát hiện một dạng hình học bất kỳ trong ảnh nhị phân đã được phát hiện biên hoặc khung xương, miễn là hình dạng ấy có thể biểu diễn được dưới dạng công thức toán học. Thường được sử dụng để phát hiện đường thẳng, đường tròn (Bradski and Kaehler, 2008).

Tuy nhiên, sau khi đã phát hiện được cơ bản các đường thẳng phân chia 9 ô cờ, do bị nhiễu nên số đường thẳng được phát hiện nhiều hơn mong muốn (nhiều hơn 4 đường), đặc điểm của các đường thẳng nằm ngang hay nằm dọc là chúng nằm gần với đường thẳng thực mà chúng tôi vẽ trên bàn cờ, do vậy, chúng tôi sử dụng giải thuật gom cụm K-means để gom các đường thẳng lân cận nhau lại thành 1 đường duy nhất.

Giải thuật gom cụm (có thể gọi là học không giám sát) (Đỗ Thanh Nghị, 2008) thực hiện việc nhóm các dữ liệu không có nhãn, sao cho các dữ liệu cùng nhóm có các tính chất tương tự nhau và dữ liệu của 2 nhóm khác nhau sẽ có các tính chất khác nhau.

Giải thuật gom cụm K-means được đề xuất bởi James MacQueen vào năm 1967.

Cho tập dữ liệu  $X = \{x_1, x_2, \dots, x_n\}$  và tập tâm  $V = \{v_1, v_2, \dots, v_k\}$ , các phần tử của tập tâm  $V$  được

chọn ngẫu nhiên không lặp lại từ tập  $X$ ,  $n$  là số điểm dữ liệu cần gom cụm,  $k$  là số cụm mà ta cần gom, với  $n \geq k$ .

Các bước thực hiện gom cụm của giải thuật K-means được thực hiện theo thứ tự như sau:

1. Tính khoảng cách giữa điểm dữ liệu và các tâm trong tập  $V$ . Tính như thế cho tất cả các điểm dữ liệu. Các khoảng cách có thể là khoảng cách Manhattan, khoảng cách Euclid,...

2. Mỗi phần tử được gán cho tâm gần nhất với nó, dựa trên khoảng cách mà ta đã tính được ở bước 1.

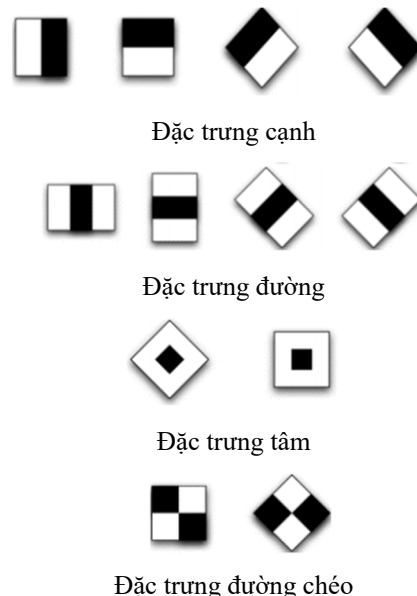
3. Cập nhật lại các tâm trong tập tâm  $V$ , giá trị của mỗi tâm sẽ là giá trị trung bình cộng (mean) của các điểm dữ liệu trong cụm của nó.

4. Lặp lại bước 1, 2, 3 cho đến khi hội tụ (không có bất kỳ sự thay đổi tâm nào trong tập tâm  $V$ ).

Do đặc điểm của bài toán nên chúng tôi sử dụng  $K = 2$  để gom các đường thẳng lân cận nhau thành 2 đường thẳng dọc và 2 đường thẳng ngang. Khoảng cách Manhattan được sử dụng để tính toán độ tương đồng.

## 2.3 Nhận dạng dấu ‘X’

Đặc trưng Haar-like (Viola and Jones, 2001) được tạo thành bằng việc kết hợp các hình chữ nhật đen, trắng với nhau theo một kích thước và trật tự nào đó (Châu Ngân Khánh và Đoàn Thanh Nghị, 2014). Một số đặc trưng Haar-like cơ bản được minh họa trong Hình 2.



**Hình 2: Một số đặc trưng Haar-like cơ bản**

Các đặc trưng Haar-like có thể có các kích thước khác nhau, bằng cách thay đổi kích thước

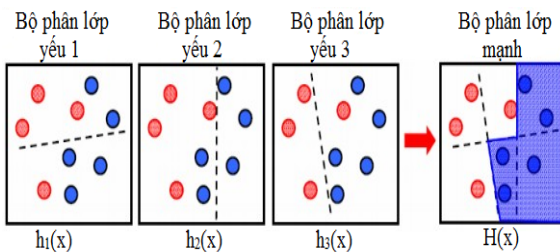
các hình chữ nhật tạo nên nó. Điều này cho phép giải thuật máy học sử dụng các đặc trưng Haar-like có thể nhận dạng các vật thể có kích thước khác nhau.

Giá trị của đặc trưng Haar-like được xác định bởi độ chênh lệch giữa tổng giá trị pixel mức xám nằm trong vùng đen so với vùng trắng hoặc ngược lại:

$$f(x) = \frac{\text{Tổng vùng đen (các mức xám của pixel)}}{\text{Tổng vùng trắng (các mức xám của pixel)}} \quad (1)$$

Đặc trưng Haar-like có thể diễn đạt được mối liên hệ giữa các vùng ảnh trong nó, điều mà bản thân từng điểm ảnh không thể hiện được.

Bằng cách thể hiện các đặc trưng bằng đặc trưng Haar-like và tính nhanh các đặc trưng thông qua ảnh tích phân thì để huấn luyện máy tính có thể nhận dạng được dấu ‘X’ của người chơi đánh vào, chúng tôi sử dụng mô hình phân lớp phân tầng với mỗi tầng là bộ phân lớp AdaBoost. Bộ phân lớp AdaBoost (Freund *et al.*, 1997), là một bộ phân lớp mạnh phi tuyến, hoạt động với nguyên tắc kết hợp tuyến tính các bộ phân lớp yếu – bộ phân lớp có tỉ lệ phân lớp đúng lớn hơn 50% (Ji and Ma, 1997) (chẳng hạn như gốc quyết định (Decision Stump), Bayes thơ ngây (Naive Bayes),...) để tạo nên một bộ phân lớp mạnh. AdaBoost sử dụng trọng số để đánh dấu các mẫu khó nhận dạng. Trong quá trình huấn luyện, cứ mỗi bộ phân lớp yếu được xây dựng thì giải thuật sẽ tiến hành cập nhật lại trọng số để chuẩn bị cho việc xây dựng bộ phân lớp tiếp theo. Cập nhật bằng cách tăng trọng số của các mẫu nhận dạng sai và giảm trọng số của các mẫu nhận dạng đúng bởi bộ phân lớp yếu vừa xây dựng. Bằng cách này, thì bộ phân lớp sau có thể tập trung vào các mẫu mà bộ phân lớp trước làm chưa tốt. Cuối cùng, các bộ phân lớp yếu sẽ được kết hợp lại tùy theo mức độ tốt của chúng để tạo nên bộ phân lớp mạnh (Hình 3).



**Hình 3: Kết hợp các bộ phân lớp yếu thành bộ phân lớp mạnh duy nhất**

Các bộ phân lớp yếu  $h_k(x)$  được lấy từ việc trích các đặc trưng Haar-like, được biểu diễn như sau:

$$h_k(x) = \begin{cases} 1 & \text{nếu } p_k f_k(x) < p_k \theta_k \\ -1 & \text{nếu ngược lại} \end{cases}$$

Trong đó:

$x$ : mẫu hay cửa sổ con cần xét ( $x = (x_1, x_2, \dots, x_n)$ ) là véc-tơ đặc trưng mẫu).

$p_k$ : hệ số chiều của bất phương trình,  $p_k \in \{1, -1\}$ .

$f_k$ : giá trị của đặc trưng Haar-like được tính như trong công thức (1).

$\theta_k$ : ngưỡng do người dùng đặt ra.

Bộ phân lớp mạnh  $H(x)$  được tạo ra bằng cách kết hợp tuyến tính  $N$  bộ phân lớp yếu  $h_k(x)$  (2). Vấn đề then chốt là thực hiện chọn lựa các trọng số  $\alpha_1, \alpha_2, \dots, \alpha_n$  thích hợp:

$$H(x) = \text{sign} \left( \sum_{k=1}^N \alpha_k h_k(x) \right)$$

Là một mô hình phân tầng với mỗi tầng là một giải thuật AdaBoost (Viola and Jones, 2002, 2001) sử dụng bộ phân lớp yếu là gốc quyết định (decision stump cây quyết định có độ sâu là 1, từ đó đưa ra kết luận chỉ dựa trên 1 thuộc tính duy nhất) với các đặc trưng Haar-like. Được sử dụng để đánh giá một vùng ảnh nào đó trong ảnh gốc có giống với vật thể đang cần phát hiện hay không. Trong mô hình phân lớp phân tầng sử dụng giải thuật AdaBoost, các bộ phân lớp được sử dụng là giải thuật AdaBoost.

## 2.4 Chơi cờ tự động

Chúng tôi sử dụng giải thuật cắt tia Alpha – Beta để tìm ra nước đi tốt nhất có thể cho máy tính, với tình hình bàn cờ hiện tại. Ý tưởng chính của giải thuật cắt tia Alpha – Beta là không khai triển (tạo nút con) các nút mà nó không thể thay đổi quyết định cuối cùng ở nút cha của nó. Phương châm của giải thuật này là “nếu biết điều đó thật sự ‘tồi tệ’ thì đừng tốn thời gian tìm hiểu nó sẽ ‘tồi tệ’ đến đâu” (nguyên văn tiếng Anh: “If you have an idea that is surely bad, don't take the time to see how truly awful it is.” -- Pat Winston).

Được thực hiện theo kiểu tìm kiếm sâu, nhưng thay vì tìm kiếm tất cả các nhánh trên cây, giải thuật này sẽ cắt bớt các nhánh không cần thiết, làm giảm số nước đi phải sinh ra và từ đó làm giảm việc định trị, do đó có thể gia tăng tốc độ tìm kiếm nước đi tốt nhất có thể.



### 3 KẾT QUẢ THỰC NGHIỆM

#### 3.1 Mô tả thực nghiệm

Chúng tôi thực nghiệm hệ thống chơi tic-tac-toe với người đầu với máy trong phiên bản thực tế tăng cường bằng ngôn ngữ lập trình C/C++, sử dụng bàn cờ được kẻ bằng bút lông trên giấy in A4 (210x297 mm), sử dụng thư viện mã nguồn mở OpenCV phiên bản 3.1.0 trên máy tính xách tay có các thông số như sau:

Tiêu chí	Giá trị	Ghi chú
Hiệu máy tính	Asus X550CC-XO055D	
Bộ xử lý (CPU)	Intel Core i5 - 3337U	
	Processor (3M Cache, 1.80Ghz)	
Bộ nhớ (RAM)	4GB DDR3/1600MHz	
Hệ điều hành	Windows 8.1 Pro 64 bit	
Webcam	Asus webcam 900KP	
Tốc độ webcam	30 khung hình / giây	

#### 3.2 Huấn luyện nhận dạng dấu 'X'

Thông số huấn luyện mô hình phân lớp phân tầng sử dụng AdaBoost, AdaBoost được chúng tôi sử dụng là Gentle AdaBoost:

“-numPos 140 -numNeg 247 -numStages 14 -maxFalseAlarmRate 0.3 -minHitRate 0.98”

Trong đó:

– *numPos 140*: số lượng mẫu dương được huấn luyện, hay số lượng ảnh có đối tượng (dấu 'X') là 140.

– *numNeg 247*: số lượng mẫu âm được huấn luyện, hay số lượng ảnh không có đối tượng (không có dấu 'X') là 247.

– *numStages 14*: số lượng tầng (giai đoạn) của mô hình phân lớp phân tầng, với mỗi tầng sử dụng bộ phân lớp AdaBoost được huấn luyện là 14.

– *maxFalseAlarmRate 0.3*: tỷ lệ nhận dạng sai đối tượng tối đa là 30%.

– *minHitRate 0.98*: tỷ lệ nhận dạng đối tượng (dấu 'X') tối thiểu là 98%.

#### 3.3 Kết quả kiểm tra

Khi kiểm thử với tập kiểm tra gồm có 113 mẫu: 63 mẫu dương (dấu 'X') và 50 mẫu âm (không phải dấu 'X'), chúng tôi thu được confusion matrix như sau với độ chính xác (accuracy) hơn 98%:

Dự đoán =>	Dương	Âm
Dương	63	0
Âm	2	48

### 4 KẾT LUẬN VÀ ĐỀ XUẤT

Chúng tôi vừa trình bày các kỹ thuật để xây dựng một hệ thống chơi tic-tac-toe với người đầu với máy trong phiên bản thực tế tăng cường. Quy định của trò chơi là người chơi sẽ là người ra nước đi đầu tiên, người chơi sẽ dùng bút lông để đánh dấu 'X' lên bàn cờ đã được chuẩn bị trước, camera của laptop thu lại hình ảnh bàn cờ, và từ hình ảnh đó, chúng tôi đề xuất áp dụng kỹ thuật khung xương hóa đối tượng ảnh, kết hợp với phép biến đổi Hough và giải thuật gom cụm K-means để có thể phát hiện được 4 đường thẳng tạo nên bàn cờ, sử dụng các hàm vẽ cơ bản trong thư viện OpenCV để vẽ các đường thẳng đề chồng lên các đường thẳng đã phát hiện được (áp dụng công nghệ thực tế tăng cường), việc phát hiện chính xác 4 đường thẳng của bàn cờ là cơ sở để chúng tôi có thể trích ra được chính xác 9 ô cờ. Để phát hiện ra dấu 'X' của người chơi, chúng tôi áp dụng mô hình phân lớp phân tầng với mỗi tầng là bộ phân lớp AdaBoost, tốc độ nhận dạng nhanh và ổn định, kết quả nhận dạng chính xác đạt hơn 98%, sau khi người chơi đã ra nước đi của mình và hệ thống đã nhận dạng được nước đi đó thì đến lượt chơi của máy tính. Để máy tính có thể đưa ra các nước đi “thông minh” nhất có thể thì chúng tôi áp dụng giải thuật cắt tia Alpha – Beta, với ưu điểm là dễ dàng cài đặt và tốc độ tính toán nhanh.

Do chương trình hiện tại còn hạn chế về sự linh động, chưa thể tự động định vị được bàn cờ nên hướng nghiên cứu tiếp theo của bài báo là phát triển module để có thể tự động định vị được bàn cờ. Với cơ sở đã xây dựng được, chúng tôi có thể phát triển các trò chơi tương tự cần sử dụng trí tuệ nhân tạo như trò giải ô số sudoku, trò cờ ca-rô, cờ vua,... Cũng với trò tic-tac-toe phiên bản thực tế tăng cường, nhưng có thể nâng cấp lên cách thức chơi mới hơn, đó là thay vì chơi bên giấy, chúng tôi có thể chiếu bàn cờ lên bảng, tường hoặc một mặt phẳng bất kỳ (tất nhiên cần phải có máy chiếu), bên cạnh việc đi nước cờ bằng bút lông, thì ta có thể dùng phấn trắng, hoặc tiện lợi hơn thì có thể dùng bàn tay để ra nước đi, chương trình sẽ ra nước đi bằng cách chiếu nước đi ấy lên mặt phẳng mà chúng tôi sử dụng.

### TÀI LIỆU THAM KHẢO

- C. Ji and S. Ma. Combinations of weak classifiers. Advances in Neural Information Processing Systems. 1997.
- Châu Ngân Khánh và Đoàn Thanh Nghị. Nhận dạng mặt người với giải thuật Haar like feature – cascade of boosted classifiers và đặc trưng SIFT. Tạp chí khoa học Trường Đại học An Giang, quyển 3 số 2: trang 15 – 24. 2014.

- Đỗ Thanh Nghị. Giải thuật gom cụm, Bài giảng môn “Khai khoáng dữ liệu”. Khoa Công nghệ thông tin – Truyền thông, Đại học Cần Thơ. 2008.
- G. Bradski and A. Kaehler. Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc. 2008.
- G. Klinker, D. Stricker and D. Reiners. An optically based direct manipulation interface for human-computer interaction in an augmented world. Virtual Environments' 99. Springer Vienna, trang 53-62. 1999.
- J. Maguire and D. Saltzman. Augmented Reality Tic-Tac-Toe. Department of Electrical Engineering. Stanford University. 2013.
- P. A. Maragos and Ronald W. Schafer. "Morphological skeleton representation and coding of binary images." IEEE Transactions on Acoustics, Speech, and Signal Processing 34.5 (1986): 1228-1244.
- P. V. C. Hough. Method and means for recognizing complex patterns. U.S. Patent No. 3,069,654. 1962.
- P. Viola and M. Jones. Fast and robust classification using asymmetric Adaboost and a detector cascade. Advances in neural information processing systems 2: trang 1311-1318. 2002.
- P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE. 2001.
- R. Kresch and D. Malah. "Skeleton-based morphological coding of binary images." IEEE Transactions on Image Processing 7.10 (1998): 1387-1399.
- Y. Freund và R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1): trang 119 – 139. 1997.