Public Interactive Display

Ke He, Richard Green
Department of Computer Science and Software Engineering, University of Canterbury

The abstract goes here.

***Index Terms*—hand tracking, public display, deep learning**

I. Introduction

WAITING for a cup of coffee to be made is a boring process, especially if you are buying one from the Reboot Cafe with not many seats available. The only entertainment you can possibly get is from the TV above the cafe, but it just shows some boring advertisements. To make the coffee waiting experience more entertaining, this paper presents a way for coffee buyers to play a classic game of snake with the public displays above cafe. The game consists of a hand tracking algorithm to track user's hands and a face detection algorithm to match the hands detected with a face.

Both the hand tracking algorithm and face detection algorithm must be capable of tracking multiple hands in a public environment with a web camera. The algorithms has to reliably track a users hand or face from a distance away while being efficient enough to process multiple frames per second. Previous solutions to this problem or a similar problem is examined and a possible solution proposed with experiments done to test the accuracy and viability of the solution.

II. Background and Related Work

Hand tracking and face recognition are the important elements of this project. This section reviews a wide range of hand tracking algorithms and several face detection algorithms.

A. Hand Tracking

Mitra *et al.* [1] have surveyed a wide range of hand detection techniques. Stastistical methods such as hidden Markov model(HMM), particle filtering and condensation algorithm and finite state machine(FSM) are based on the probability of an object being a hand given the previous locations and properties said object was in. These approaches have been proven successful, however it requires a large amount of computation power, which is hard to achieve real time hand detection.

Ghotkar *et al.* [2] investigated hand segmentation with color sampling. This paper shows the choice of color space is of utter importance to the performance of segmentation algorithm. The most successful method proposed in the paper involves separating hand color from the background with HSV color space, using Canny edge detection algorithm to detect edges of the hand and finally a edge traversal algorithm to eliminate background edges detected.

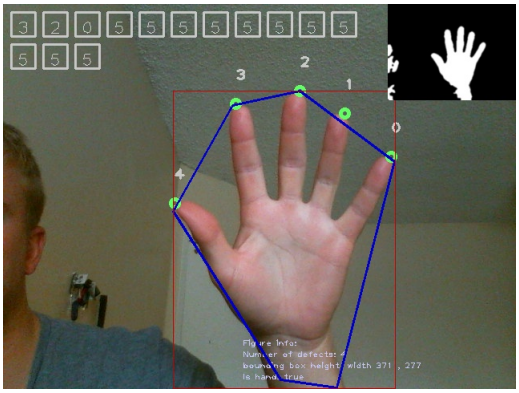

Fig. 1. Result of the hand detection algorithm [4]

Wilson [3] have investigated the hand recognition algorithm proposed by Andresen [4], using hand color sampling. The algorithm is initialized by sampling colors from multiple regions of a hand, then the hand is extracted from the background using the sampled color and turned into a binary image. Median blur is applied to smooth and eliminate noise in the binary image. The hand is then processed by applying a convex contour onto the image and filter out irrelevant defects. This algorithm is able to detect the hand gesture and finger position accurately as shown in figure one.

Other hand detection algorithms requires additional hardware. Wang *et al.* [5] presented a way of real-time hand tracking and detection using a color glove. Participants need to put on a color glove in order for the program to recognize hand position and gesture. This method provides a quick and accurate tracking of the hand, but the hardware limitations makes it impractical.

Conventional hand detection algorithms are extremely limited. Statistical methods such as HMM, particle filtering and FSM are computationally heavy; color sampling methods are extremely sensitive to background noise and requires color sampling of the hand; hardware assisted hand detection overcomes problems with previous methods, however in a public environment it is difficult to secure the hardware.

B. Face Recognition

The face detection framework developed by Viola *et al.* [6] was capable of rapidly processing images at high accuracy. Unlike previous attempts which focuses on pixels, this framework focuses on features. It is able to do so by using a new way

of representing image called integral image. Integral image represents an pixel (x,y) by sum all pixels above and to the left of (x,y), this representation allows entire features to be classified by the AdaBoost classifier. The AdaBoost classifier was trained to detect small number of important features to speed up the classification process. Moreover, the framework is able to cascade multiple complex classifiers in order to focus on more promising areas of the image, which further boosts the speed of the face detection. The idea of feature classification and cascading feature structure is very similar to the current stat-of-the-art object classification algorithm using deep convolutional nets.

Neural networks have been in literature for a long time. The first working multi-layer perceptron was developed by Ivakhnenko *et al.* [7] in 1965. However due to the large amount of computations required by neural networks, it was generally ignored by the computer vision society. Over the years CPUs and GPUs have grown more powerful, along with a improved method of learning, deep learning.

Deep learning is a subset of representation learning where raw data are fed into the algorithm, and the algorithm is capable to automatically discover features or representation that is needed for detection and classification. Deep learning methods are able to do so by having multiple levels of representations. Each representation level is obtained by transforming the representation of the previous level into a more abstract representation starting from the raw data. Multiple layers of representation with parameter sharing have greatly reduced the memory requirements [8].

Deep learning is so power it is not limited to face recognition. ImageNet Large Scale Visual Recognition Competition [9] is a benchmark in object classification. Competitors are faced with challenges to categorize millions of pictures into hundreds of categories. Winners of each year's competition presents new architectures to improve the speed and accuracy of object classification. Notable winners include AlexNet [10], VGG net [11] and ResNet [12].

III. Proposed Solution

Traditional hand recognition algorithms are extremely limited. Statistical methods using HMM, particle filtering and FSM are computationally heavy; color sampling methods are extremely sensitive to background lighting and occlusions and requires a hand sampling phase[13]; hardware assisted methods are infeasible in a public environment. A new family of algorithms in object recognition is deep learning and it has shown competitive performance compared to traditional object detection algorithms.

Deep learning have made major break-through across multiple domains such as image recognition, speech recognition, natural language processing and many more [8]. It is more robust in recognizing multiple objects, a downside of is that the algorithm cannot detect fine features such as finger positions. Nevertheless, this disadvantage is negligible, since hand position is of importance and hand gestures are not useful.

Multiple hand detection packages using tensorflow framework has already been developed [13] [14], this project will be built on top of Dibia's [13] framework. The face detection module will be based on facenet [15]. A multi-threaded video input is used to further speed up this process [16].

To match a hand with a face, a simple hand matching algorithm is used. The two closest hands within a certain x range of the head are considered, to make it more robust, the valid range will vary depending on the size of the head detected. An assumptions to be made for this algorithm to be successful is that other coffee buyers are not going to invade your personal bubble.

The game will be a classic game of snake. To start the game, users have to raise their hand above the head for 3 seconds. The snake will be following the user's hand at all times, if the user's hand stops moving when the snake reaches the hand, it will continue to move forward. In order for the user to move hand around the entire screen, an invisible, scaled down version of the screen will be formed around the user's hand. User's hand in this space will be projected onto the screen. If the hand moves over the boundaries of the screen, it will be capped. The user can pause the game by raising two hands above the head.

IV. Conclusion

The conclusion goes here.

References

[1] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, May 2007, ISSN: 1094-6977. DOI: 10.1109/TSMCC.2007.893280.

[2] A. S. Ghotkar and G. K. Kharate, "Hand segmentation techniques to hand gesture recognition for natural human computer interaction," *International Journal of Human Computer Interaction (IJHCI)*, vol. 3, no. 1, p. 15, 2012.

[3] A. Wilson, "Interactive public display: Upper body pose detection of multiple subjects," Manuscript submitted for publication, 2017.

[4] M. S. Simen Andresen Vegar Osthus. (2013). Hand tracking and recognition with opencv, [Online]. Available: http://simena86.github.io/blog/2013/08/12/hand-tracking-and-recognition-with-opencv/.

[5] R. Y. Wang and J. Popovic, "Real-time hand-tracking with a color glove," in *ACM transactions on graphics (TOG)*, ACM, vol. 28, 2009, p. 63.

[6] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004, ISSN: 1573-1405. DOI: 10 . 1023 / B : VISI . 0000013087 . 49260 . fb. [Online]. Available: https://doi.org/10.1023/B:VISI.0000013087.49260.fb.

[7] A. Ivakhnenko and V. Lapa, *Cybernetic Predicting Devices*, ser. Jprs report. CCM Information Corporation, 1973. [Online]. Available: https://books.google.co.nz/books?id=FhwVNQAACAAJ.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014. arXiv: 1409.0575. [Online]. Available: http://arxiv.org/abs/1409.0575.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385.

[13] D. Victor, *Real-time hand tracking using ssd on tensorflow*, https://github.com/victordibia/handtracking, 2017.

[14] L. Marie, *Hands detection*, https://github.com/loicmarie/hands-detection, 2017.

[15] D. Sandberg, *Facenet*, https://github.com/davidsandberg/facenet, 2017.

[16] A. Rosebrock, *Faster video file fps with cv2.videocapture and opencv*, https://www.pyimagesearch.com/2017/02/06/faster-video-file-fps-with-cv2-videocapture-and-opencv/, 2017.