Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 9, No. 2, 2021, 227-234.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

A Novel Approach to Communicate with Video Game Character using Cascade Classifiers

Mahmood Mohamadzadeh and Hossein Khosravi*

Faculty of Electrical Engineering, Shahrood University of Technology, Shahrood, Iran.

Article Info

Article History:
Received 15 June 2020
Revised 13 September 2020
Accepted 24 October 2020

DOI: 10.22044/jadm.2020.9788.2110

Keywords:
Interactive Game, Mobile Game, Hand Gestures Recognition, Face Detection, Cascade Classifiers.

**Corresponding author: hosseinkhosravi@shahroodut.ac.ir(H. Khosravi).*

Abstract

Today, video games have a special place among entertainment. In this work, we have developed an interactive video game for mobile devices. In this game, the user can control the game's character by his face and hand gestures. Cascading classifiers along with the Haar-like features and local binary patterns are used for hand gesture recognition and face detection. The game's character moves according to the current hand and face state received from the frontal camera. Various ideas are used to achieve the appropriate accuracy and speed. Unity 3D and OpenCV for Unity are employed in order to design and implement the video game. This game is written in C#, and developed for both the Windows and Android operating systems. The experiments show an accuracy of 86.4% in the detection of five gestures. It also has an acceptable frame rate, and can run at 11 fps and 8 fps in Windows and Android, respectively.

1. Introduction

Video games are among the most popular entertainments, and are growing day by day. An interesting idea for today's games is to control them by the user body interaction. For example, in PlayStation and Xbox consoles, the users can control the character of some games with their hands and head. Such games are a fun sport, especially in today's life, where many people have little physical activity and mostly sit at the table for long hours. Commercial game consoles are expensive and only can be used with a monitor or TV. On the other hand, the new generation of children and adolescents are more interested in mobile games. They spend most of their time with their smartphones or tablets. If mobile games can communicate with the user gestures, they will help our kids and teens to practice physical activities while using their smartphones, and this will make them healthier.

In this research work, we intend to design a game that the users can play with their mobile phones and transfer all commands to the game environment through the movements of their heads and hands. In this game, the character in the game has to go through a path that includes various obstacles and reach the end. In order to play this game, we are required to have an Android smartphone or a Windows device that has a frontal camera. The game has a 3D graphics environment, and is developed in Unity 3D.

The rest of this paper is organized as follows. The related research works are reviewed in Section 2. Section 3 presents the details of the proposed algorithm based on cascade classifiers. The experimental results and comparison with other works are discussed in Section 4. Finally, the conclusion is presented in Section 5.

2. Related Works

Various articles have been published in the field of face and hand gesture recognition that have been implemented on various platforms. Sarkar *et al*. [1] have used a deep neural network to identify faces using a mobile phone. Features have been extracted from the outputs of the first 5 layers of a pre-trained AlexNet network. Sliding windows of different sizes are employed in order to identify

faces in different scales. This method is very slow and cannot be used for the real-time applications.
Chen *et al*. [2] have used the Haar-like features for hand gesture recognition. In this article, four hand gestures have been identified using the Viola-Jones method. Positive samples are provided by a single person with 450 samples for each hand movement. Also 500 images that do not include hand movements are selected for negative examples. This method is claimed to be real-time but the results obtained show the opposite. It requires about 300 ms for a single posture recognition in a frame. Since they recognize four postures, the total required time is more than 1 s, which is definitely not real-time.

Khodabakhsh *et al*. [3] have classified hand movements using the image processing techniques. These gestures include numbers 1 to 5, punch, confirmation, top, bottom, right, and left. They change the color space of the image into the YCbCr space. This is performed due to the importance of colors in hand detection. In the next step, the hand area is located through the background subtraction and the color segmentation method. Then they find the smaller parts of the hand. This method can process 2 frames per second, which is not suitable for real-time applications.

A group of articles in this field have designed a video game using the Kinect sensor to control the game's character [4-6]. Qingtang *et al*. [4] have developed an interactive game in which the speed of the character decreases whenever the user bends backward and increases whenever he bends forward. They used the Kinect SDK for gesture recognition, and did not propose a novel algorithm. Zhu *et al*. [5] have proposed a hand gesture detection system that can track the movements of both hands and identify their gestures to control a video game. In this research work, some regions of interest have been proposed to reduce the processing time. Farzin Nezhad *et al*. [6] have designed a sports game that uses the Kinect sensor to perform the right ping pong exercises without the presence of a coach. This game has been designed using the Unity software, and no explicit novelty has been proposed.

Elrefaei *et al*. [7] have developed a real-time face recognition system for criminal detection. The police use a smartphone camera to record a video of the suspect. The detection and tracking operations are performed on the user side. The video frames including the detected and tracked faces are then sent to a server. There is a face recognition system on the server based on the Viola-Jones algorithm [8]. The identified person's personal information is then sent to the police smartphone.

Mao *et al*. [9] have used deep networks in order to identify objects so that the network can be run for embedded applications in real-time. They developed a lightweight network based on YOLOv3 called Mini-YOLOv3. The proposed network is twice as fast as YOLOv3, while its accuracy does not differ significantly. In another research work, Nanda [10] has also developed a system based on YOLOv3 for sign language translation. Although YOLOv3 has a very good performance for object detection and classification, it has a very low speed, especially when dealing with mobile devices. Fang *et al*. [11] have tried to improve the speed and accuracy of YOLOv3-tiny to be suitable for the low-power devices. In order to achieve an efficient object detection model for environments with limitations, with the origin of YOLOv3-tiny, the Tinier-YOLO model is suggested in this paper. This model has lower parameters, while it has a better accuracy and is more suitable for real-time applications.

Karabasi *et al*. [12] have proposed hand gesture recognition for deaf-mute communication based on mobile phones, which can translate sign language using HSV color space. Lahiani *et al*. [13] have proposed a hand gesture recognition system for Android devices. They have used color-based hand segmentation and median filter for smoothing, and finally, a Support Vector Machine has been employed as the classifier.

In [14], the YUV color space was used with the help of the CAMShift algorithm to distinguish between the background and skin color, and the naïve Bayes classifier was implemented to assist with gesture recognition. The proposed system faces some challenges such as illumination variation, where light changes affect the result of the skin segment. Other challenges are the degree of gesture freedom that affect directly the output result by rotation change.

Mao, Mazzia, Fang, and their colleagues [9, 11, 15] have used deep neural networks in their works, and have achieved great results on Nvidia graphic cards that support the CUDA framework. These methods are almost real-time on these GPUs; however, they cannot be used in embedded devices that usually do not have such graphic cards.

Rautaray *et al*. [16] have proposed a hand gesture recognition system including detection, tracking, and recognition module. The detection and tracking part of the proposed algorithm is separated. They have also used background subtraction, which will cause the system to crash if the user holds his hand steady.

Hosseini *et al*. [17] have proposed a hand gesture recognition for sign language. They used the mean-shift algorithm to track and background subtraction to separate moving parts. They also used thresholds in order to find the hand area. The use of background subtraction and thresholding has improved the execution speed but has reduced accuracy.

Among all the methods used for gesture recognition, algorithms based on Viola-Jones require fewer computations, and are a better choice for real-time applications. Therefore, these methods are still popular. In this work, we also used several cascade classifiers based on the Viola-Jones algorithm. The details will be presented in the next section.

3. Proposed Algorithm

3.1 Viola-Jones Classifier

The Viola-Jones-based object detection is among the first object detection algorithms that could be used in the real-time applications. Some specifications such as simplicity, reliability, and implementation in image processing libraries like OpenCV have made it quite popular, and today, it is widely used for face detection [18-21].
The Viola-Jones algorithm contains three key components [8]:

- Integral images for rapid feature computation.
- Adaboost to create strong classifiers (also known as feature selection).
- Combining the selected features into a cascade structure to get a cascade classifier.

It is the cascade structure that makes the classifier extremely rapid. The cascade structure allows vast regions of the image to be discarded with a minimum effort, focusing a heavier computation on the most promising regions. This is why the Viola-Jones framework is still popular. You can attach any feature descriptor into the Viola-Jones framework or use any form of AdaBoost while the approach remains the same.

Another advantage of this algorithm compared to the convolutional neural networks or deep networks is the amount of space required to run it. Given that the convolutional neural networks require multiple pass-throughs, and therefore, store large amounts of information, CNNs require much more space locally than the Viola-Jones algorithm. This becomes important when we are looking to implement the face or gesture detection algorithms in mobile devices, where the storage capacity is more limited than a computer.

Based on the above, we also use the Viola-Jones algorithm for our gesture recognition tasks.

3.2 Feature Extraction

The character in the video game is controlled by the user's face and three hand movements. Also the user's face image must be in front. The user's hand gestures include:

- Palm up
- Palm to the right
- Palm to the left

Two different types of features are extracted from images including the Haar-like and LBP[1] features. Both are fed into the Viola-Jones cascade classifier. They differ in the training time, accuracy, and processing time. The LBP-based cascade classifier is somewhat faster but is less accurate than the Haar-like features.

The function of LBP [22] is to label the pixels of an image by comparing the central pixel of each window against its 3×3 neighbors and assigning 0 or 1 based on this comparison. Finally, the binary code of the neighboring values is saved as a feature (Figure 1). These numbers are then represented as window representation, and a histogram is generated.

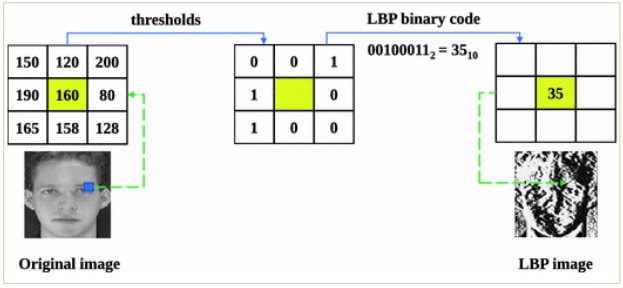

Figure 1. Algorithm of the local binary pattern.

The Haar-like features use rectangular features that are computed from two, three, and four adjacent rectangles. In the case of two rectangles, the feature is computed as the difference between the sum of the pixels in the two rectangles. The features associated with three and four rectangles are calculated similarly (Figure 2). The sum of the pixels in the white rectangles is subtracted from the sum of the pixels in the red rectangles.

[1] Local Binary Patterns

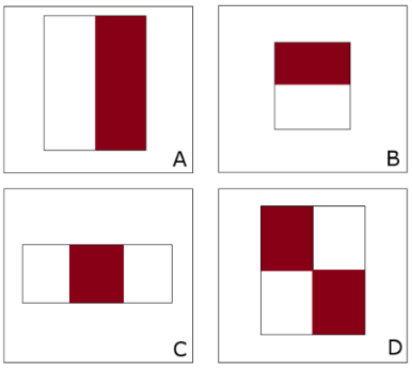

Figure 2. Four sets of rectangles in the Haar-like feature extraction.

3.3 Face Detection and Gesture Recognition

Since in our interactive game, the player that controls the game character is near the frontal camera, its face is almost big and clean. Thus, we perform face detection using the LBP feature, which is faster than the Haar-like features. For this, the whole image is scanned at different scales, and face candidates are found and the best face is selected as the final face.

Having the face detected, it is time to recognize the hand gestures. In this step, we used the following idea to reduce the calculations:

When playing a video game, the user must be in front of the mobile phone and play the game with his face and hand movements. Naturally, the hands are always located at the bottom of the face and on both sides. We used this idea to reduce computations. After detecting the face position, we define two ROIs[1] in the image according to the face location. Now, hand gesture recognition is performed only in these ROIs, as shown with red boxes in Figure 3.

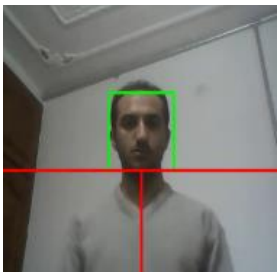

Figure 3. ROIs for detecting hand gestures (red boxes).

We trained four cascading classifiers in this work:

- Face detector
- Palm up detector (Palm-up)
- Palm to the right detector (Palm-right)
- Palm to the left detector (Palm-left)

[1] Region of Interest

The training process uses a large number of positive and negative images as well as two types of features. These parameters along with two parameters used for the test are shown in Table 1. In this table, the *number of stages* is related to the number of cascade stages in the training phase. The *Min neighbor* parameter specifies how many neighbors each candidate rectangle should have to retain during the detection phase. The scale factor parameter specifies how much the image size is reduced at each image scale. Positive images are those images that contain the object of interest, e.g. palm-right, and negative images are samples that do not include the object of interest.

Table 1. The training and test parameters of the

Classifier	Face	Palm-up	Palm-left	Palm-right
No. of features	139	420	335	606
No. of stages	20	20	18	20
Feature Type	LBP	Haar	Haar	Haar
No. of negative images	1500	5000	5000	5000
No. of positive images	3000	2000	2000	2000
Sample size	24×24	24×24	24×24	24×24
Min Neighbor (test)	2	2	2	2
Scale factor (test)	1.1	1.1	1.1	1.1

Implementation of cascade classifiers in OpenCV allows us to train a multi-class classifier to recognize all classes simultaneously. However, our experiment shows that in this way, the cascade classifier slows down. In other words, the fewer classes in the classifier, the less time it would take to process an image and perform classification. Thus, we decided to train separate classifiers for our classification tasks.

Suppose that the time required to process a frame in a three-class classifier is 3 times the processing time of a frame in a single-class classifier:

- Prediction time of a single-class classifier: t
- Prediction time of a three-class classifier: $3t$

For the multi-class classifier, the gesture recognition will be as shown in Figure 4. As one can see here, after generating ROIs, the hand gesture is detected in the right area. This process will take 3 units of time (3t). However, if no gesture is found in the right area, the search is continued in the left area, and it will take the same time (3t) that means the maximum required time is 6 units of time ($2 \times 3t = 6t$). On average, a hand gesture can be recognized in 4.5t in this scenario.

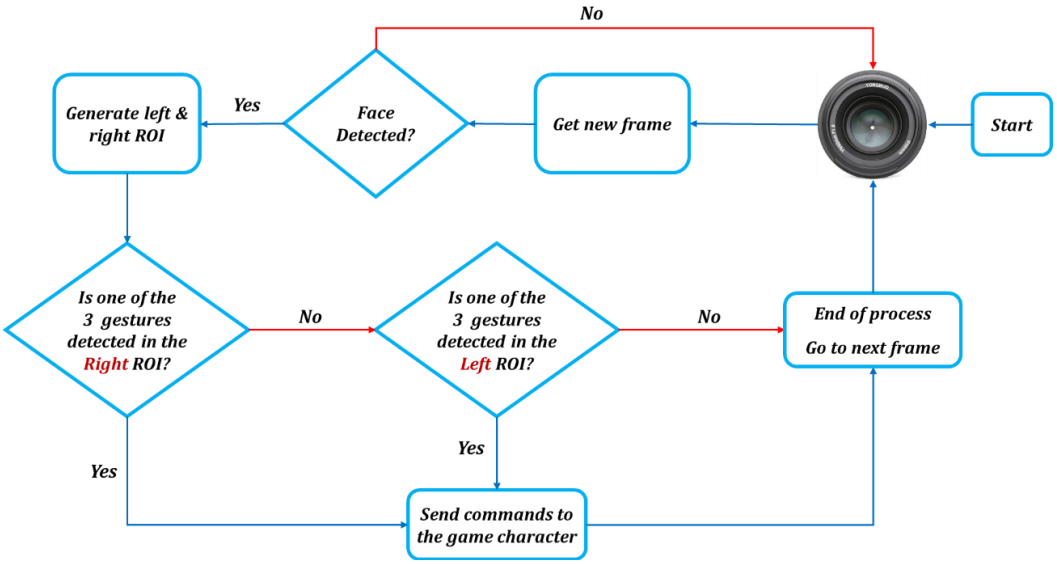

Figure 4. A flowchart for the hand gesture recognition using a multi-class classifier.

We decided to use single-class classifiers and changed the process as shown in Figure 5. In this scenario, at first, we try to find the hand gesture *palm-up* in the *right* ROI. This process will take one unit of time (t). If this gesture is not found in the *right* ROI, we try to find *palm-right* in this area. This process again takes (t) time. Next, we search for *palm-up* and *palm-left* in the *left* area, each taking another (t) time.

In this way, the maximum processing time is when the user shows a *palm-left* gesture and it is 4 × t because it will be found after four trials. On average, a hand gesture can be found in about 3.3 units of time, which is better than that of the multi-class classifier. Our experiments also approved this fact.

Table 2 shows the maximum and average times required in the single-class and multi-class scenario.

After recognizing the user's face and hand gestures, their equivalent movements will be sent to the game character. These movements are as follow:

- User's *face* detected: *Move forward*
- Simultaneous detection of the *face* and the *palm-up* gesture: *Jump*
- Simultaneous detection of the *face* and the *palm-right* gesture: *Turn right*
- Simultaneous detection of the *face* and the *palm-left* gesture: *Turn left*

Table 2. Units of time required for hand gesture recognition in multi-class and single-class classifiers.

Hand gesture	Single-class classifier	Multi-class classifier
Palm-up in the *right* ROI	1×t	3×t
Palm-right in the *right* ROI	2×t	3×t
Palm-up in the *left* ROI	3×t	6×t
Palm-left in the *left* ROI	4×t	6×t
Average time required for hand gesture recognition	3.3×t	4.5×t

4. Experimental Results

In this section, the accuracy and speed of the proposed method of gesture recognition are presented and compared with the other works. It must be mentioned that we developed two versions of this game, one for Android and another for the Windows operating system. This is an advantage of the Unity 3D gaming engine that allows game development for different operating systems.

4.1. Gesture Recognition Results

In order to evaluate the accuracy of our gesture recognition method, we conducted an experiment in which the user played the game and performed each gesture 50 times during the game. The results obtained are shown in Table 3. The palm-up gesture was performed using both the right and left hands so it occurred 100 times rather than 50.

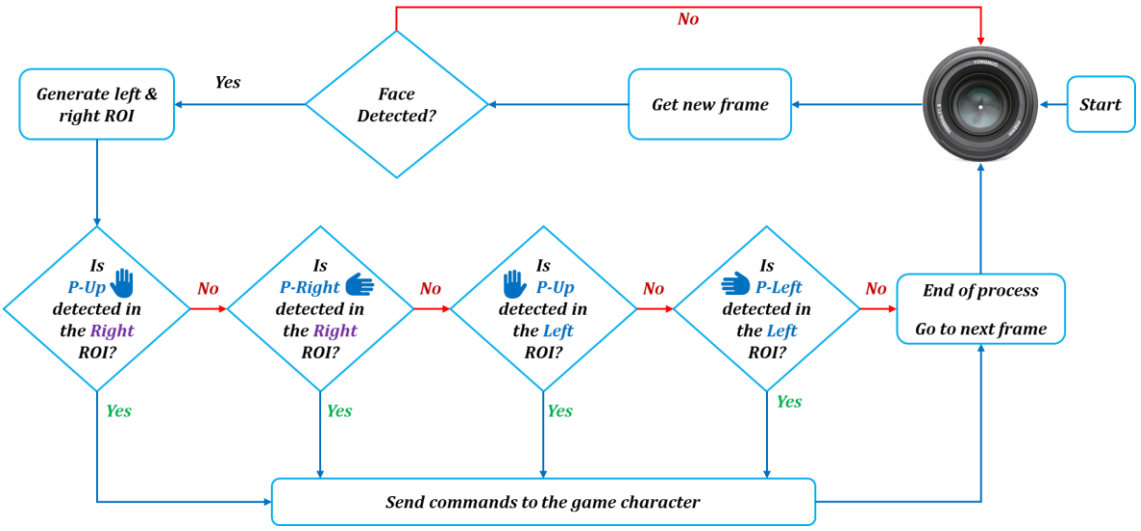

Figure 5. The proposed flowchart for the hand gesture recognition using single-class classifiers for hand gestures.

Table 3. Evaluating the accuracy of the gesture recognition on the mobile phone.

Classifier	No. of moves	Identified moves	Missed moves	Acc (%)
Palm-up	100	80	20	80
Palm-right	50	43	7	86
Palm-left	50	43	7	86
Face detection	50	50	0	100
Total	250	216	34	86.4

As one can see in Table 3, the accuracy of face detection is 100%. This is due to more positive samples used for training the face cascade classifier. Furthermore, the face has a better texture than the hand, and can be found more easily. It also is different from hand gestures, and cannot be confused with them.

Within hand gestures, *palm-up* has a lower accuracy. This is because we use a single classifier for palm-up in both hands, while *palm-left* and *palm-right* only happen in their respective hands.

In order to compare the results of our method against the others, two similar works [2, 3] were considered. The results obtained are shown in Figure 6. As it can be seen, the accuracy of the proposed method is comparable with these works and acceptable. The accuracy and speed are two key parameters in the classification systems. However, in this work, the speed of the algorithm was more important than accuracy, and we tried to develop an algorithm that could run on mobile devices. Thus, the accuracy of 86.4% was acceptable for this work. It must be mentioned that the proposed method was evaluated on two different devices, a mobile phone with Android and a PC with Windows OS, and the accuracy was the same.

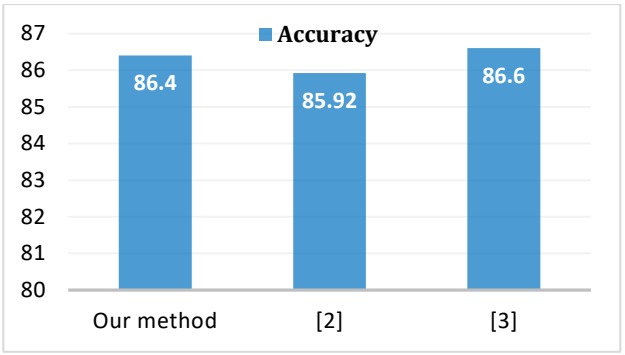

Figure 6. Comparing the accuracy of the proposed method with similar works.

4.2. Evaluating Speed of Method

We used two devices for speed measurement. One was a low-range personal computer with Core i3 CPU, 4GB of RAM, and Windows OS. Another device was a Samsung Galaxy Note 9 mobile phone with Android OS.

The developed game ran almost fluent on PC and achieved the speed of 11 frames per second. In the Android version, we achieved 8 frames per second, which was slower than the PC version. In order to compare the proposed method with the other works, we used the PC version since there was no similar work in Android OS. The results obtained are shown in Figure 7. The speed of this video game was higher compared to the other works. Meanwhile, the other works only process hand gestures but in our method, there is a computational burden according to the running video game. The other methods have a slower execution time even compared with the mobile version of this method, which has 8 FPS.

Figure 8 shows the 3D game environment and the detection of the face and two hand-gestures (*palm-left* & *palm-right*). It can be seen that the game character operates correctly according to the user gestures.

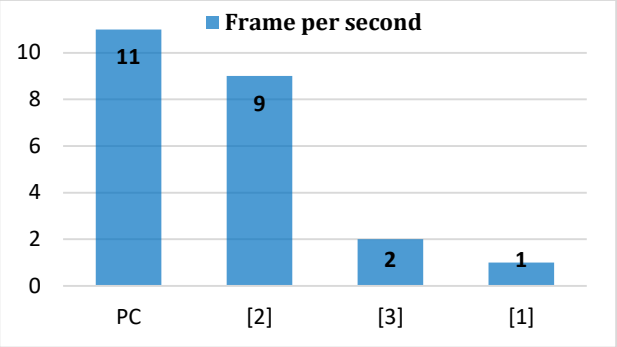

Figure 7. Comparison of the running speed.

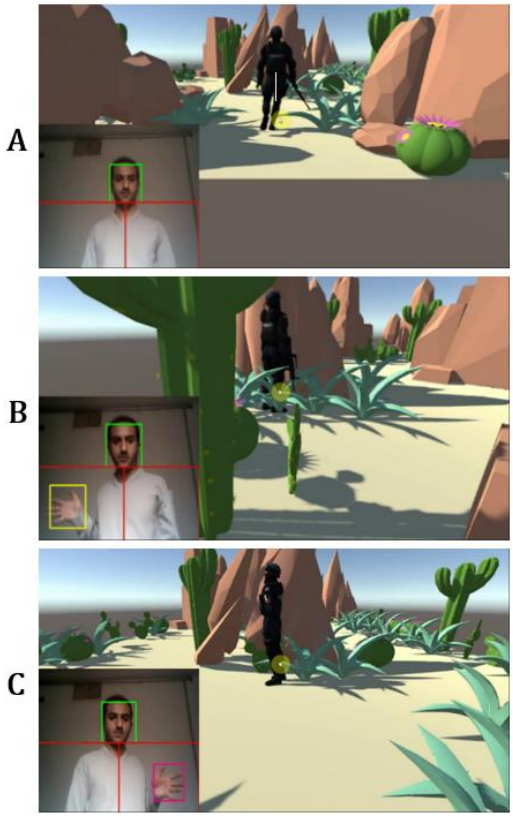

Figure 8. Playing of the interactive game by the user A) *Face detection* makes the character to run forward (B) *Palm-right* gesture makes the character turn right (C) *Palm-left* gesture makes the character turn left.

5. Conclusion

In this work, we developed an interactive video game for the Windows and Android operating systems. A new idea was used to communicate with the game's character using face detection and hand gestures. Four cascade classifiers were trained to detect appropriate gestures using the LBP and Haar-like features. The proposed method showed an accuracy of 86.4% in the gesture detection and the speed of 11 fps and 8 fps on the PC and mobile phone, respectively.

Among the advantages of this research work were its good execution speed and acceptable accuracy. The graphic design of the video game was at a higher level compared to the similar works, and had an adverse effect on the speed of gesture recognition.

As the future work, we can add a tracking module to speed-up the process of gesture recognition and have a more stable system. Also for a precise control of the game character, the user's body skeleton can be calculated, and the user's joints can be connected to the game character so that s/he can have a complete control over the game.

References

[1] Sarkar S, P.V., and Chellappa R. "Deep Feature-based Face Detection on Mobile Devices". in *2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*. Sendai, Japan: IEEE. 2016.

[2] Chen Q, G.N., and Petriu E. "Real-time Vision-based Hand Gesture Recognition Using Haar-like Features". in *2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007*. Warsaw, Poland: IEEE. pp. 1-6, 2007.

[3] Khodabakhsh, V. and M. Islami. "Detection of hand movements for human-computer interaction using image processing techniques (in Farsi)". in *Second International Conference of Researchers in Computer Engineering and Information Technology*. Tehran, Iran. 2017.

[4] Qingtang L, Y.W., Linjing W, Jingxiu H, and Peng W. "Design and Implementation of a Serious Game Based on Kinect". in *2015 International Conference of Educational Innovation through Technology (EITT)*. Wuhan, China: IEEE. pp. 13-18, 2015.

[5] Zhu Y, a.Y.B. "Real-Time Hand Gesture Recognition with Kinect for Playing Racing Video Games". in *2014 International Joint Conference on Neural Networks (IJCNN)*. Beijing, China: IEEE. pp. 3240-3246, 2014.

[6] Farzin Nezhad, F. and J. Rasti. "A computer-sports game for practicing forehand drive shading using Kinect Xbox technology (in Farsi)". in *First International Conference on Opportunities and Challenges Computer Games*. Isfahan, Iran. 2017.

[7] Elrefaei L, A.A., Alamoudi H, Almutairi S, and Al-rammah F. "Real-time face detection and tracking on mobile phones for criminal detection". in *2017 2nd International Conference on Anti-Cyber Crimes (ICACC)*. Abha, Saudi Arabia: IEEE. 2017.

[8] Viola, P. and M. Jones. "Rapid object detection using a boosted cascade of simple features". in *Proceedings of the 2001 IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition. CVPR 2001. IEEE. pp. 511-518, 2001.

[9] Mao Q, S.H., Liu Y, and Jia R, "Mini-Yolov3: Real-Time Object Detector for Embedded Applications". *IEEE Access*. Vol. 7, pp. 133529 - 133538, 2019.

[10] Nanda, M., *You Only Gesture Once (YouGo): American Sign Language Translation using YOLOv3*. 2020, Purdue University Graduate School.

[11] Fang W, W.L., and Ren P, "Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments". *IEEE Access*. Vol. 8, pp. 1935-1944, 2020.

[12] Karabasi, M., Z. Bhatti, and A. Shah. "A model for real-time recognition and textual representation of malaysian sign language through image processing". in *Proceedings of the 2013 International Conference on Advanced Computer Science Applications and Technologies*. Kuching, Malaysia. pp. 195–200, 2013

[13] Lahiani, H., M. Elleuch, and M. Kherallah. "Real time hand gesture recognition system for android devices". in *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*. IEEE. pp. 591-596, 2015.

[14] Prakash, J.G., U.K., "Hand Gesture Recognition". *International Journal of Recent Technology and Engineering*. **7**, pp. 54-59, 2019.

[15] Mazzia V, K.A., Salvetti F, and Chiaberge M, "Real-Time Apple Detection System using Embedded Systems With Hardware Accelerators: An Edge Ai Application". *IEEE Access*. Vol. 8, pp. 9102 - 9114, 2020.

[16] Rautaray, S.S. and A. Agrawal. "Interaction with virtual game through hand gesture recognition". in *2011 International Conference on Multimedia, Signal Processing and Communication Technologies*. Aligarh, India: IEEE. pp. 244-247, 2011.

[17] Hosseini, M.M. and J. Hassanian, "Applying mean shift and motion detection approaches to hand tracking in sign language". *Journal of AI and Data Mining*. Vol. 2, No. 1, pp. 15-24, 2014.

[18] Anitha, J., G. Mani, and K.V. Rao, *Driver drowsiness detection using viola jones algorithm*, in *Smart Intelligent Computing and Applications*. 2020, Springer. p. 583-592.

[19] Barnouti, N.H., et al., "Face detection and recognition using Viola-Jones with PCA-LDA and square euclidean distance". Vol. 7, No. 5, pp. 371-377, 2016.

[20] Ren, J., N. Kehtarnavaz, and L. Estevez. "Real-time optimization of Viola-Jones face detection for mobile platforms". in *2008 IEEE Dallas Circuits and Systems Workshop: System-on-Chip-Design, Applications, Integration, and Software*. IEEE. pp. 1-4, 2008.

[21] Vikram, K. and S. Padmavathi. "Facial parts detection using Viola Jones algorithm". in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE. pp. 1-4, 2017.

[22] Ojala, T., M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions". *Pattern Recognition*. Vol. 29, No. 1, pp. 51-59, 1996.

رویکرد جدید برای ارتباط با شخصیت بازی ویدیویی با استفاده از طبقه‌بند آبشاری

محمود محمدزاده و حسین خسروی*

دانشکده مهندسی برق، دانشگاه صنعتی شاهرود، شاهرود، ایران.

ارسال ۲۰۲۰/۰۶/۱۵؛ بازنگری ۲۰۲۰/۰۹/۱۳؛ پذیرش ۲۰۲۰/۱۰/۲۴

چکیده:

امروزه بازی‌های ویدیویی در بین سرگرمی‌ها جایگاه ویژه‌ای دارند. در این مقاله، ما یک بازی ویدیویی تعاملی را برای دستگاه‌های تلفن همراه توسعه داده‌ایم. در این بازی، کاربر می‌تواند شخصیت بازی را با حرکات صورت و دست خود کنترل کند. از طبقه‌بند های آبشاری با ویژگی‌های شبه هار و الگوهای باینری محلی برای تشخیص حرکات دست و تشخیص چهره استفاده می‌شود. شخصیت بازی با توجه به وضعیت فعلی دست و صورت مبتنی بر تصاویر دریافتی از دوربین سلفی، حرکت می‌کند. از ایده‌های مختلفی برای دستیابی به دقت و سرعت مناسب استفاده کرده‌ایم. به منظور طراحی و پیاده سازی بازی نرم افزارهای Unity 3D و OpenCV for Unity به کار گرفته شده‌اند. این بازی به زبان سی شارپ نوشته شده و برای هر دو سیستم عامل ویندوز و اندروید توسعه یافته است. نتایج آزمایش‌ها، دقت ٪۸۶٫۴ را در تشخیص پنج حرکت نشان می‌دهد. همچنین روش پیشنهادی نرخ فریم قابل قبولی داشته و می‌تواند با سرعت ۱۱ فریم بر ثانیه و ۸ فریم بر ثانیه به ترتیب در ویندوز و اندروید اجرا شود.

کلمات کلیدی: بازی تعاملی، بازی موبایل، تشخیص حرکات دست، تشخیص چهره، طبقه‌بند آبشاری.