Mechanics of Advanced and Smart Materials Journal
http://masm.araku.ac.ir

Implementing a Virtual Piano Using Image Processing

Tannaneh Ghadimian [a], Hassan Moradzadeh [a*]

[a] Department of Mechanical Engineering, Faculty of Engineering, Arak University, Arak, Iran

Original Article

Use your device to scan and read the article online

Citation: Ghadimian T, Moradzadeh H. Implementing a Virtual Piano Using Image Processing. Mechanics of Advanced and Smart Materials,2025;5(2):292-304.

https://10.66224/masm.5.2.292.

KEYWORDS

Virtual Piano, Image Processing,
Hand Gesture Recognition,
MATLAB,
OpenCV.

ABSTRACT

The objective of this study is to provide a method for real-time hand movement detection using only a webcam and machine vision technology, which involves image processing that can recognize multiple gestures for interaction with a computer. The functionality of this study is similar to simulating a virtual piano using hand gesture detection and specific movement detection for each piano note. The implementation was done in MATLAB and Visual Studio C++ using the OpenCV library, and a comparison between the two implementations was performed. The results show that the implementation using the OpenCV library is faster and more efficient in most cases compared to MATLAB. The accuracy of the implementation using MATLAB is 86.45%, and using OpenCV is 92.7%. In terms of time, the detection of each hand gesture corresponds to a musical note in the MATLAB environment, which takes approximately 1.39 seconds, and in the OpenCV environment, it takes approximately 1.19 seconds.

Extended Abstract

1. Introduction

Importance of human-computer interaction is increasing significantly. Creating a virtual interaction device, such as a virtual keyboard, using a webcam and machine vision techniques can be an alternative to touch screens. In this study, hand gesture tracking for virtual keyboard application using a standard webcam was designed and implemented. The

Goal of this research is to develop a program for object tracking for interaction with a computer and to create a device for virtual human-computer interaction. There are two ways to implement a virtual piano: using data glove methods and machine vision methods. Data glove methods use sensors that detect hand movements and connect those movements to a computer. Data gloves are commonly used in virtual reality (VR) applications, where the user sees a virtual image of the data glove and can manipulate virtual environment movements using the glove. Additional sensors make it easier to detect hand position and movements. However, these devices are expensive [1,2]. In contrast, machine vision methods only require a single camera and are therefore easier and less expensive to use [3]. In this study, computer vision is used to receive video from a webcam and analyze it frame by frame to detect hand movements in each frame. Techniques such as image processing are used to detect hand position and blue labels on the fingernails (as Figure 1) for processing the hand position and detecting a particular hand gesture.

* Corresponding author. Tel.:+989126466681
E-mail address: h-moradzadeh@araku.ac.ir
DOI: https://10.66224/masm.5.2.292.
Received: June 14,2024; Accepted: July 24,2024.

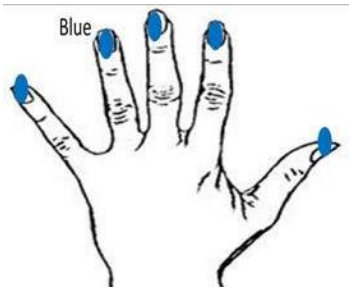


Figure 1. Blue Markers on Fingers

The input is an RGB image, so the image is divided into three color spaces: "red, green, and blue". There are three color thresholds from 0 to 1. As mentioned, blue labels were used to detect the hand to eliminate the red and green colors; the threshold for these two colors is saturated. The virtual piano of the current work was implemented in both MATLAB and Microsoft Visual Studio using the OpenCV library due to its advantages in real-time image processing. These two were then compared, and their problems were discussed. The software must meet the real-time working conditions. This is measured by the number of frames per second (fps), which provides information about the refresh rate. If the refresh rate is long, there is a delay between the actual event (musical note here) and its recognition. The software was first implemented using MATLAB and then using the OpenCV library to understand and compare the results.

2. System Design

The system uses a webcam or external camera that must be perpendicular to the hand and also uses a template like a keyboard. The output of the camera will be displayed on the monitor. The computer analyzes the images of the keyboard and markers by taking pictures of the colored fingers and detects the hand gestures.

This system is designed to recognize hand gestures using a camera and image processing techniques. The camera captures images of the hand and the keyboard, and the computer analyzes these images to detect the hand gestures. As of Figure 2, the system uses a template like a keyboard to recognize the hand gestures and display the output on the monitor.

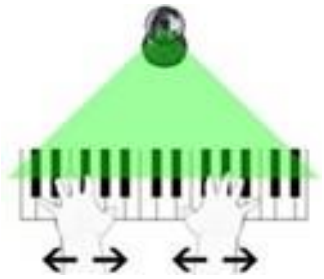


Figure 2. The Input Structure

The flowchart of the system, as shown in Figure 3, is designed as follows:

- **Image Capture:** The camera captures images, which are then processed in real-time. This process typically involves receiving images from a source that automatically captures images and then stores them frame by frame.
- **Blue Color Detection:** After storing the frames, the blue color must be detected. This is a crucial step in the process, as it helps identify the hand gestures and distinguish them from other colors.
- **Binary Image Conversion:** The captured image is then converted into a binary image, where each pixel can have only two values or levels. This step is essential for the image processing algorithms to work effectively.
- **Morphological Processing:** Some morphological processing techniques are applied to the binary image. This step helps to enhance the image quality and improve the accuracy of the hand gesture detection.

- **Center of the Image Detection:** The system then calculates the center of the identified image and draws a rectangle around it. The result is displayed to the user.

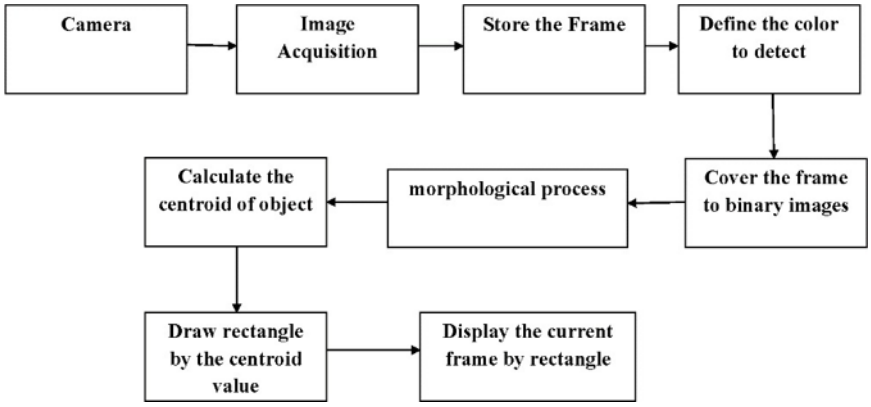


Figure 3. Color Recognition Flowchart

This flowchart outlines the key steps involved in the virtual keyboard system, from image capture to image processing and finally to the display of the recognized hand gestures.

3. Virtual Piano Implementation

The process of object tracking involves estimating the path of an object on an image during movement in a scene. In other words, a tracker assigns fixed labels to tracked objects in different frames of a video. Additionally, depending on the scope of tracking, a tracker can also provide information about the object such as direction, area, or shape of an object [5]. Object tracking methods include point tracking [6], kernel tracking [7], and silhouette tracking [8]. In this study, the Kernel tracking method was used. This model is widely used due to its relative simplicity and low computational cost.

As in Figure 4, blue color on the fingernails was used as a marker, and black and white patterns on the paper were used as a virtual keyboard.

Figure 4. Colored-Finger Interaction with Virtual Piano

After obtaining the image, various image processing techniques can be applied to the image for performing various visual tasks. The Blob algorithm [9] is used to detect regions such as corners and edges, and then the desired color range is determined, which is inactive for red and green colors and active for blue colors. After finding the blue color, the frame must be converted to binary and then scaled to grayscale, and then the morphologic processing is performed, and finally, the center of the object is calculated. To reduce the salt and pepper noise, a median filter is used. Finally, the color separation techniques are used. After separating the colors and detecting the blue color, a rectangle is drawn [10].

The users place both hands under the camera. They can execute the program and play a virtual piano that includes the notes Do, Re, Mi, Fa, Sol, La, Si, Do#, Re#, Mi#, Fa#, Sol#, La#. Depending on the hand gesture, the corresponding note will be created, and it will remain until another correct movement occurs. This way, the user can control the note length, allowing them to produce notes in a fascinating way to create music similar to a real piano. Figure 5, for example, shows the hand gestures representing the notes Do and Re.

Figure 5. Hand Gestures for Piano Notes Do and Re

4. Experimental Results

To test the system's performance, 6 random users were defined, and each note was played by each user 40 times, resulting in a total of 480 records. Table 1 compares the results of these experiments for 40 repetitions and for different users using MATLAB and OpenCV.

Table 1. Time needed for Recognizing 40 Notes (in Seconds)

User #	1	2	3	4	5	6	Time (s)
MATLAB	53.52	51.2	55.52	56.96	57.6	58.72	55.6
OpenCV	48.4	44.8	50.8	46.1	42.3	52.6	47.5

According to the values in the Table 1, it takes approximately 1.39 seconds on average in the MATLAB environment and 1.19 seconds in the OpenCV environment to detect each note. Table 2 shows the accuracy of playing notes in MATLAB and OpenCV environments.

Table 2. Accuracy and Errors

Environment	False Detection Rate	Missed Stroke Rate	Incorrect Detection Rate	Accuracy
MATLAB	1.29%	3%	9.37%	86.45%
OpenCV	1.37%	1.33%	4.62%	92.7%

False detection occurs when a key is not pressed but a character is output. The missing stroke occurs when a key is pressed but no character is output. Incorrect detection is equivalent to announcing the wrong character instead of another character. It is observed that implementation in the OpenCV environment leads to fewer errors and better accuracy

5. Conclusion

A virtual piano was implemented using MATLAB and OpenCV. Generally, OpenCV is the most comprehensive open-source library for machine vision and has a large user community. Also, it has more functions for machine vision than MATLAB, and many of its functions are implemented on GPU. Additionally, OpenCV C++ code runs faster than MATLAB code. The results show the performance in two ways. The accuracy of the implementation using MATLAB is 86.4%, while using OpenCV results in an accuracy of 92.7%. In terms of speed, both implementations were performed on a personal computer. Detection of each hand gesture in the MATLAB environment needs approximately 1.39 seconds, and in the OpenCV environment, this was about 1.19 seconds. If implemented with embedded hardware such as ARM microcontrollers and FPGA ICs and also using parallel processing techniques and concurrency, it is possible to achieve detection times in fractions of a second, providing a better experience for the musician and music listener.

مکانیک مواد پیشرفته و هوشمند، دوره ۵ شماره ۲ سال ۱۴۰۴ صفحات ۲۹۲ تا ۳۰۴

مکانیک مواد پیشرفته و هوشمند

شاپا: ۴۲۲۰-۲۷۸۳

مجله علمی مکانیک مواد پیشرفته و هوشمند

طناز قدیمیان[الف]، حسن مرادزاده[ب]*

[الف] کارشناسی ارشد مکاترونیک، دانشکده فنی و مهندسی، دانشگاه اراک، اراک، ایران، hmoradzadeh@gmail.com

[ب] استادیار، گروه مهندسی برق، دانشکده فنی و مهندسی، دانشگاه اراک، اراک، ایران، h-moradzadeh@araku.ac.ir

واژگان کلیدی

صفحه‌کلید مجازی، پردازش تصویر، تشخیص ژست دست، MATLAB، OpenCV.

تاریخ دریافت:۱۴۰۳/۰۳/۲۵

تاریخ پذیرش: ۱۴۰۳/۰۴/۰۴

چکیده

هدف از این مطالعه، ارائه‌ی رویکردی برای تشخیص حرکات دست در زمان واقعی است که تنها از یک وب‌کم و نیز فناوری بینایی ماشین استفاده می‌کند و در حقیقت پردازش تصویری است که می‌تواند چندین ژست را برای استفاده در تعامل با رایانه تشخیص دهد. کارکرد این مطالعه نیز شبیه‌سازی یک پیانوی مجازی با استفاده از تشخیص ژست دست و تشخیص حرکات خاص برای هر نت پیانو است. پیاده‌سازی در MATLAB و C++ Studio Visual با استفاده از کتابخانه‌ی OpenCV انجام و مقایسه بین دو پیاده‌سازی صورت گرفته است. نتایج نشان می‌دهد پیاده‌سازی با استفاده از کتابخانه‌ی OpenCV سریع‌تر و در بیشتر موارد کارآمدتر از MATLAB است. دقت پیاده‌سازی با استفاده از MATLAB برابر با ۸۶/۴۵ درصد و با استفاده از OpenCV برابر با ۹۲/۷ درصد است. از نظر زمانی نیز تشخیص هر ژست دست متناظر با یک نت موسیقی در محیط MATLAB به زمانی حدود ۱/۳۹ ثانیه و با کتابخانه‌ی OpenCV به زمانی حدود ۱/۱۹ ثانیه نیاز دارد.

۱- مقدمه

امروزه رشد فناوری کامپیوتر همچنان ادامه دارد و اهمیت تعامل انسان و کامپیوتر به‌شدت در حال افزایش است. ایجاد یک دستگاه تعامل مجازی انسان با رایانه مانند صفحه‌کلید مجازی با استفاده از دوربین و تکنیک‌های بینایی ماشین می‌تواند یک راه جایگزین برای صفحه‌نمایش لمسی باشد. در این مطالعه، ردیابی ژست انگشت‌ها برای کاربرد کیبورد مجازی با استفاده از یک وب‌کم معمولی طراحی و پیاده‌سازی شده است. هدف این پژوهش، ایجاد یک برنامه‌ی ردیابی اشیاء برای تعامل با رایانه و نیز توسعه‌ی دستگاهی جهت تعامل مجازی رایانه و انسان است. دو راه برای پیاده‌سازی پیانوی مجازی وجود دارد: استفاده از روش‌های مبتنی بر دستکش‌های داده[1] و روش‌های مبتنی بر بینایی ماشین. روش‌های مبتنی بر دستکش‌های داده از حسگرهایی استفاده می‌کنند که حرکت دست‌ها را حس کرده و آن حرکات را با رایانه ارتباط می‌دهد. دستکش‌های داده معمولاً در کاربردهای واقعیت مجازی[2] یا VR استفاده می‌شود که در آن کاربر تصویری از دستکش داده را می‌بیند و می‌تواند حرکات محیط مجازی را با استفاده از دستکش دست‌کاری کند. وجود سنسورهای اضافی، تشخیص موقعیت و حرکات دست را آسان می‌سازد. با این حال، این دستگاه‌ها گران‌قیمت هستند [۱و۲]. در مقابل، روش‌های مبتنی بر بینایی ماشین فقط به یک دوربین نیاز دارند و لذا استفاده از آنها آسان‌تر و ارزان‌تر است [۳].

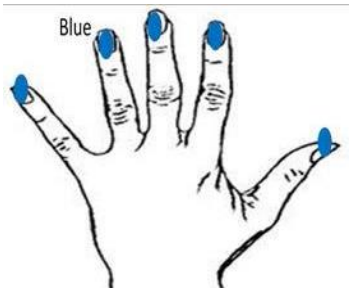
[1] Data-glove
[2] Virtual Reality

* نویسنده مسئول؛ تلفن: ۰۹۸۹۱۲۶۴۶۶۶۸۱+

آدرس پست الکترونیک: h-moradzadeh@araku.ac.ir

۲-

در این مطالعه، از بینایی کامپیوتری استفاده می‌شود که ویدیو را از یک وب‌کم دریافت و آن را فریم‌به‌فریم تجزیه و تحلیل می‌کند تا حرکات دست را در هر فریم تشخیص دهد. از تکنیک‌های پردازش تصویر برای تشخیص موقعیت دست و برچسب‌های آبی روی ناخن‌ها همانند شکل ۱ برای پردازش موقعیت دست استفاده شده است.

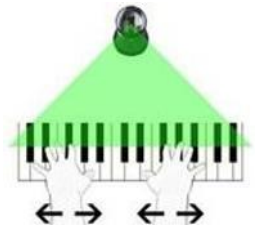


شکل ۱ نشانگرهای رنگی روی ناخن‌ها

تصویر ورودی یک تصویر RGB است، بنابراین تصویر به سه فضای رنگی "قرمز، سبز و آبی" تقسیم می‌شود. سه آستانه‌ی رنگ از ۰ تا ۱ وجود دارد. همان‌طور که ذکر گردید از برچسب‌های آبی برای تشخیص دست استفاده شد تا رنگ قرمز و سبز حذف گردد؛ آستانه‌ی این دو رنگ اشباع شده است. پیانوی مجازی کار حاضر در دو محیط MATLAB و نیز Microsoft Visual Studio با استفاده از کتابخانه‌ی OpenCV به‌دلیل مزایای آن در پردازش تصویر بلادرنگ پیاده‌سازی شد. سپس این دو با هم مقایسه شده و در مورد مشکلات آنها بحث می‌شود. نرم‌افزار باید شرایط کار بلادرنگ را برآورده کند. این امر از طریق تعداد فریم بر ثانیه یا fps[1] اندازه‌گیری می‌شود که اطلاعاتی در مورد نرخ تازه‌سازی ارائه می‌دهد. اگر نرخ تازه‌سازی[2] طولانی باشد، بین رویداد واقعی و شناسایی آن تاخیر وجود دارد. نرم‌افزار ابتدا با استفاده از MATLAB و سپس با استفاده از کتابخانه‌ی OpenCV برای درک و مقایسه‌ی نتایج بین آنها پیاده‌سازی شد.

۳-

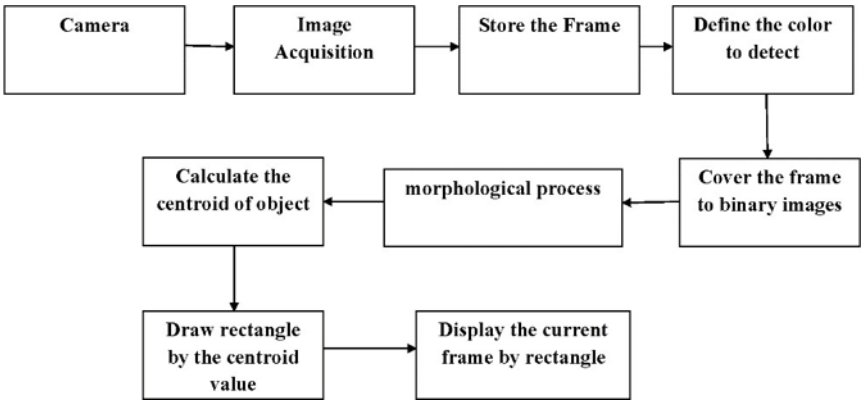
همانند شکل ۲، سیستم از یک وب‌کم یا دوربین اکسترنال استفاده می‌کند که باید عمود بر دست باشد و نیز از الگویی مانند صفحه‌کلید استفاده می‌کند. خروجی دوربین روی مانیتور نشان داده خواهد شد. کامپیوتر انگشتان رنگی را با گرفتن تصاویر صفحه‌کلید و نشانگرها آنالیز نموده و تشخیص می‌دهد.



فلوچارت کاری سیستم مطابق شکل ۳ طراحی شده است. در ابتدا دوربین عکس‌ها را گرفته و اطلاعات تصاویر اخذ می‌شود. فرایند اخذ تصویر در زمان واقعی معمولاً شامل دریافت تصاویر از منبعی است که به‌طور خودکار تصاویر را گرفته و سپس فریم‌به‌فریم ذخیره می‌کند. پس از ذخیره‌ی فریم‌ها، رنگ آبی باید تشخیص داده شود. سپس تصویر باید به تصویر باینری

[1] frames per second
[2] refresh rate

تبدیل شود که در آن برای هر پیکسل فقط دو مقدار یا دو سطح ممکن وجود دارد. در مرحله‌ی بعد برخی فرآیندهای ریخت‌شناسی[1] باید اعمال شود. سیستم سپس مرکز تصویر شناسایی‌شده را محاسبه و یک مستطیل دور آن رسم نموده و نتیجه را نمایش می‌دهد [۴].



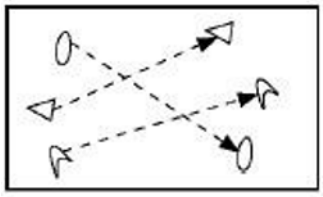
٤-

٤-١-

ردیابی اشیاء فرآیند تخمین مسیر حرکت یک جسم در صفحه‌ی تصویر در حین حرکت در یک صحنه است. به‌عبارت دیگر، یک ردیاب برچسب‌های ثابتی را به اشیاء ردیابی‌شده در فریم‌های مختلف یک ویدیو اختصاص می‌دهد. علاوه بر این، بسته به دامنه‌ی ردیابی، یک ردیاب همچنین می‌تواند اطلاعات حول شیء مانند جهت، ناحیه، یا شکل یک شیء را ارائه دهد [۵].

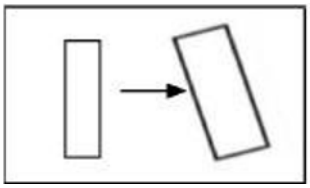
٤-٢-

ردیابی نقطه‌ای[2]: همانند شکل ۴، اشیاء شناسایی‌شده در فریم‌های متوالی با نقاطی نشان داده می‌شود و ارتباط نقاط براساس حالت قبلی شیء است که می‌تواند شامل موقعیت و حرکت جسم باشد. این رویکرد به یک مکانیسم خارجی برای شناسایی اشیاء در هر فریم نیاز دارد [۶].

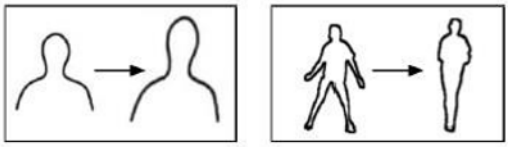


ردیابی هسته‌ای[3]: هسته به شکل و ظاهر اجسام اشاره دارد. به‌عنوان‌مثال، هسته می‌تواند یک الگوی مستطیلی یا یک شکل بیضوی با یک هیستوگرام مرتبط باشد. اشیاء با محاسبه‌ی حرکت هسته در فریم‌های متوالی ردیابی می‌شوند. این حرکت معمولاً به‌صورت تبدیلات پارامتری مانند تبدیل[4]، چرخش[5] و وابستگی[6] می‌باشد. در شکل ۵ شمایی از ردیابی کرنل یا هسته‌ای دیده می‌شود [۷].

[1] Morphological processes
[2] Point tracking
[3] Kernel tracking
[4] Translation
[5] Rotation
[6] Affine



ردیابی سیلوئت (شبح‌ها)[1]: این ردیابی با تخمین ناحیه‌ی شیء در هر فریم انجام می‌شود. این روش‌های ردیابی از اطلاعات رمزگذاری‌شده در ناحیه‌ی اطراف شیء استفاده می‌کنند. این اطلاعات می‌تواند به‌صورت مدل‌های تراکم و شکل ظاهری همچون شکل ۶ باشد که معمولاً به‌صورت نقشه‌های لبه[2] هستند. با توجه به مدل‌های داده‌شده برای اشیاء، شبح‌ها با تطبیق شکل[3] یا تکامل خطوط[4] ردیابی می‌شوند [۸].



در این مطالعه از ردیابی هسته‌ای استفاده شد. این مدل به‌دلیل سادگی نسبی و هزینه‌ی محاسباتی کم به‌طور گسترده مورد استفاده قرار گرفته است.

٤-۳-

در این پژوهش از رنگ آبی بر روی ناخن انگشت به‌عنوان علامت[5] و نیز از طرح‌های سیاه و سفید روی صفحه‌ی کاغذ به‌عنوان صفحه‌ی پیانوی مجازی استفاده شد (شکل ۷).

پس از به‌دست‌آمدن تصویر، روش‌های مختلف پردازش را می‌توان بر روی تصویر برای انجام بسیاری از وظایف بینایی مختلف اعمال کرد. از الگوریتم Blob برای تشخیص مناطقی مانند گوشه[6] و لبه[7] استفاده شده و سپس محدوده‌ی رنگ موردنظر تعیین شده است که برای رنگ‌های قرمز و سبز غیرفعال و برای رنگ آبی فعال است [۹]. پس از یافتن رنگ آبی، کادر باید به تصویر باینری و سپس مقیاس خاکستری[8] تبدیل شود و سپس فرآیند ریخت‌شناسی انجام و نهایتاً مرکز جسم محاسبه شود. به‌منظور

[1] Silhouette tracking
[2] Edge maps
[3] Shape matching
[4] Contour evolution
[5] Marker
[6] Corner detection
[7] Edge detection
[8] Gray scale

کاهش نویز نمک و فلفل[۱] از فیلتر میانه[۲] استفاده شده است. برای تشخیص طیف رنگ‌ها از تفکیک رنگ[۳] استفاده شده است. پس از جداسازی رنگ‌ها و تشخیص رنگ آبی، یک مستطیل رسم می شود [۱۰].

کاربر هر دو دست را زیر دوربین قرار می‌دهد. او می‌تواند برنامه را اجرا کرده و پیانوی مجازی را بنوازد که شامل نت‌های DO, Re, Mi, Fa, Sol, La, Si, Do#, Re#, Mi#, Fa#, Sol#, La# است.

بسته به ژست دست[۴]، نت متناظر ایجاد خواهد شد و تا زمانی که حرکت صحیح دیگری رخ ندهد، نواخت این نت باقی می ماند. به‌این‌شکل می‌توان طول نت را کنترل نمود که به کاربر اجازه می‌دهد نت‌ها را به روشی جالب تولید کند تا موسیقی حاصل مانند پیانوی واقعی باشد. انواع ژست‌های دست که بیانگر نت‌های مختلف است در شکل‌های ۸ تا ۱۴ دیده می‌شود.

Re , Do

شکل ۹ ژست دست‌ها برای نت‌های ۲ و ۳ پیانو Mi, Fa

شکل ۱۰ ژست دست‌ها برای نت‌های ۴ و ۵ پیانو Sol, La

شکل ۱۱ ژست دست‌ها برای نت‌های ۶ و۷ پیانو Si, Do#

شکل ۱۲ ژست دست‌ها برای نت‌های ۸ و ۹ پیانو Re#, Mi#

[1] Salt and pepper noise
[2] Median filter
[3] Color separation
[4] Hand gesture

Fa#, Sol# ۱۰

#La

۵-

جهت آزمایش نحوه‌ی کارکرد، ابتدا ۶ کاربر تصادفی تعریف شده و هر نت توسط هر کاربر ۴۰ بار نواخته شده و در مجموع ۴۸۰ رکورد جمع‌آوری می‌شود. جدول ۱ مقایسه‌ی نتایج زمانی این آزمایش‌ها را برای ٤٠ بار تکرار و به ازای کاربرهای مختلف با استفاده از MATLAB و OpenCV نشان می‌دهد.

کاربر	۱	۲	۳	۴	۵	۶	میانگین زمان تشخیص ۴۰ کاراکتر
MATLAB	۵۳/۵۲	۵۱/۲	۵۵/۵۲	۵۶/۹۶	۵۷/۶	۵۸/۷۲	۵۵/۶
OpenCV	۴۸/۴	۴۴/۸	۵۰/۸	۴۶/۱	۴۲/۳	۵۲/۶	۴۷/۵

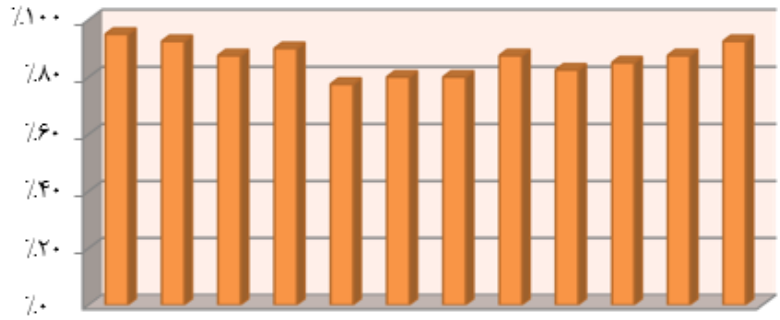
مطابق مقادیر جدول فوق، در محیط MATLAB به‌طور میانگین به ۱/۳۹ ثانیه و در محیط OpenCV به ۱/۱۹ ثانیه زمان برای تشخیص هر نت نیاز است. جدول ۲ دقت نواختن نت‌ها را در محیط‌های MATLAB و OpenCV نشان می‌دهد

جدول ۲ میزان دقت و خطای دو سیستم

MATLAB	۱/۲۹٪.	۳٪.	۹/۳۷٪.	۸۶/۴۵٪.
OpenCV	۱/۳۷٪.	۱/۳۳٪.	۴.۶۲٪.	۹۲/۷٪.

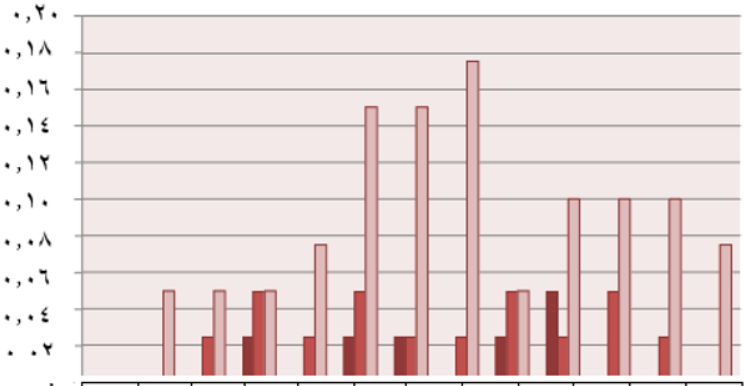
تشخیص کاذب[1] زمانی اتفاق می‌افتد که کلیدی فشار داده نشده اما یک کاراکتر صادر می‌شود. ضربه‌ی از دست‌رفته[2] زمانی رخ می‌دهد که یک کلید فشار داده شده اما هیچ کاراکتری صادر نشود. تشخیص نادرست[3] معادل اعلام اشتباه یک کاراکتر به‌جای کاراکتری دیگر است. دیده می‌شود که پیاده‌سازی در محیط OpenCV به خطای کمتر و نیز دقت بهتر منجر می‌شود. نتایج کار در شکل‌های ۱۵ تا ۱۸ آمده که به‌ترتیب نشان‌دهنده‌ی دقت و میزان خطای سیستم‌های پیاده‌شده با MATLAB و OpenCV است.

[1] False detection
[2] Missed stroke
[3] Incorrect detection



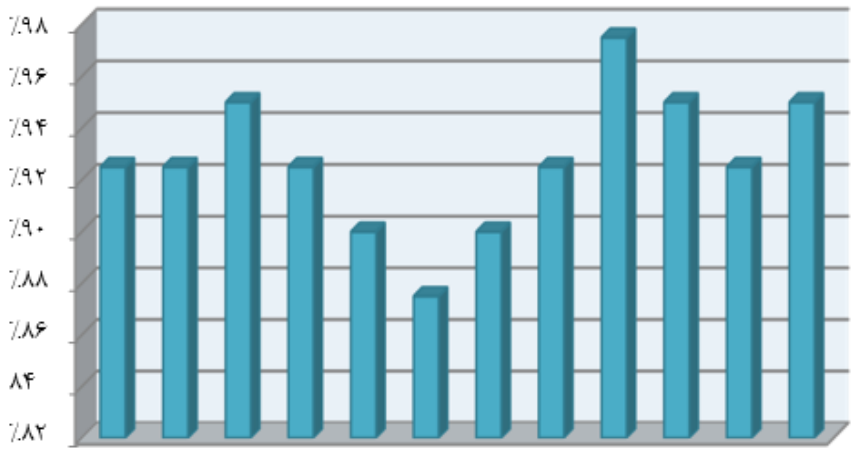
	Do	Re	Mi	Fa	Sol	La	Si	C#	D#	F#	G#	A#
درصد	۹۵	۹۲/۵	۸۷/۵	۹۰	۷۷/۵	۸۰	۸۰	۸۷/۵	۸۲/۵	۸۵	۸۷/۵	۹۲/۵

MATLAB



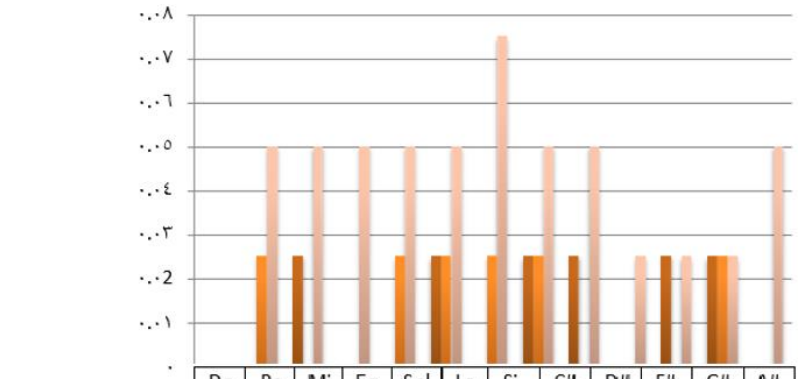
	Do	Re	Mi	Fa	Sol	La	Si	C#	D#	F#	G#	A#
False detection rate	۰	۰	۲.۵	۰	۲.۵	۲.۵	۰	۲.۵	۵	۰	۰	۰
Missed strokes	۰	۲.۵	۵	۲.۵	۵	۲.۵	۲.۵	۵	۲.۵	۵	۲.۵	۰
Incorrect detection	۵	۵	۵	۷.۵	۱۵	۱۵	۱۷	۵	۱۰	۱۰	۱۰	۷.۵

MATLAB



	Do	Re	Mi	Fa	Sol	La	Si	C#	D#	F#	G#	A#
درصد	۹۳	۹۲/۵	۹۵	۹۳	۹۰	۸۸	۹۰	۹۲/۵	۹۷.۵	۹۵	۹۲/۵	۹۵

OpenCV



	Do	Re	Mi	Fa	Sol	La	Si	C#	D#	F#	G#	A#
False detection	۰	۲.۵	۰	۰	۲.۵	۰	۲.۵	۲.۵	۰	۲.۵	۲.۵	۰
Missed strokes	۲.۵	۰	۰	۲.۵	۲.۵	۲.۵	۲.۵	۰	۰	۰	۲.۵	۰
Incorrect detection	۵	۵	۵	۵	۵	۷.۵	۵	۵	۲.۵	۲.۵	۲.۵	۵

OpenCV

۶-

در این مطالعه یک پیانوی مجازی با استفاده از MATLAB و OpenCV پیاده‌سازی شد. به‌طورکلی OpenCV جامع‌ترین کتابخانه‌ی منبع باز برای بینایی ماشینی و نیز دارای جامعه‌ی کاربری بزرگی است ونیز عملگرهای بیشتری نسبت به MATLAB برای بینایی ماشین دارد و بسیاری از توابع آن نیز بر روی GPU پیاده‌سازی می‌شود. عمدتاً کد OpenCV C++ سریعتر از کد MATLAB اجرا می‌شود چرا که زبان MATLAB یک زبان برنامه‌نویسی مفسری است و این امر بر عملکرد آن از نظر سرعت تأثیر منفی می‌گذارد. همچنین برای کاربردهای زیادی طراحی شده است که با جعبه ابزارهای متعدد آن از مالی گرفته تا ابزارهای تخصصی تجزیه‌وتحلیل DNA نشان داده می‌شود. درحالی‌که کتابخانه‌ی OpenCV صرفاً و به‌طور تخصصی برای پردازش تصویر ساخته شده است. نتایج به دو صورت عملکرد را نشان می‌دهد. دقت پیاده‌سازی با استفاده از MATLAB برابر با ۸۶.٤ درصد بوده ولی استفاده از OpenCV به دقت ۹۲/۷ درصد منجر می‌شود. از نظر سرعت نیز هر دو نوع پیاده‌سازی با کامپیوتر شخصی صورت گرفته است. در صورت پیاده‌سازی با سخت‌افزارهای نهفته همچون میکروکنترلرهای ARM و قطعات FPGA و استفاده از تکنیک‌های پردازش موازی و نیز همزمانی می‌توان به زمان‌های تشخیص کسری از ثانیه و لذا تجربه‌ی بهتر برای نوازنده و شنونده‌ی قطعات موسیقی رسید. ضمناً از یافته‌های این تحقیق می‌توان در کاربردهای جالب دیگر نیز استفاده نمود. به‌عنوان مثال در [۱۱] از بررسی سرعت تایپ کاربر روی صفحه‌کلید کامپیوتر جهت تشخیص زودهنگام بیماری پارکینسون بهره گرفته شده است. تلفیق یافته‌های پژوهش جاری با کار تشخیصی مذکور می‌تواند تجربه‌ای دلپذیرتر برای بیمار و پزشک به‌همراه داشته باشد.

Authorship Contribution Statement

Tannaneh Ghadimian

Biography: MSc student in Computer engineering, Arak University.

Contribution Statement: Methodology, Software Implementation, Validation, Writing – original draft, Writing – review & editing.

Dr. Hassan Moradzadeh

Biography: Assistant Professor of Electrical Engineering at Arak University.

Contribution Statement: Contribution Statement: Conceptualization, Supervision, Project administration, Investigation, Writing – review & editing.

۷- مراجع

[1] Ji B, et al. Flexible strain sensor-based data glove for gesture interaction in the metaverse: A review. Int J Hum Comput Interact. 2023;40(4):1–20.

[2] Lin BS, Lee IJ, Yang SY, Lo YC. Design of an inertial-sensor-based data glove for hand function evaluation. Sensors. 2018;18(5):1545.

[3] Kverh B, Lipanje M, Batagelj B, Solina F. Piano Crossing—Walking on a keyboard. Acta Graph. 2010;21(3–4):223–24.

[4] Yoshinari T, Tsutomu T, Shojiro N. Design and implementation of a real-time fingering detection system for piano performance. In: Proceedings of the International Computer Music Conference (ICMC); 2006:67–74.

[5] Sakthivel S, Anbarasi A. Hand gesture recognition system for human-computer interaction. Int J Recent Trends Electr Electron Eng. 2014:121–32.

[6] Zhou X, Koltun V, Krähenbühl P. Tracking objects as points. In: Proceedings of the European Conference on Computer Vision (ECCV); 2020:474–90.

[7] Shen C, Kim J, Wang H. Generalized kernel-based visual tracking. IEEE Trans Circuits Syst Video Technol. 2010;20(1):119–30.

[8] Jiang X, Sun J, Ding H, Li C. A silhouette-based novel algorithm for object detection and tracking using information fusion of video frames. Cluster Comput. 2019;22(1):391–8.

[9] Petrovic VL, Popovic-Bozovic JS. Towards real-time blob detection in large images with reduced memory cost. In: Proceedings of the 3rd International Conference on Electrical, Electronic and Computing Engineering (IcETRAN); 2016.

[10] Samantaray A, Nayak SK, Mishra AK. Hand gesture recognition using computer vision. Int J Sci Eng Res. 2013;4(6):1602–9.

[11] Sarhangi A, Nezhad A. Parkinson disease recognition based on type speed on keyboard. In: Proceedings of the Civilica Conference; 2023. (In Persian).