ROCK-PAPER-SCISSORS PROGRAM

Yash Varde, Karen Reynaga

**OVERVIEW**

In this interactive Rock, Paper, Scissors program, the user takes a photograph of their hand gesture, which will be one of the following: rock, paper, or scissors. The program, using computer vision and OpenCV methods, identifies the player’s choice. The program also randomly selects its move from the three options. The winner of the game is then determined by comparing the player’s choice and the program’s choice, following the traditional rules of the game. The user can use the webcam or provide an image of their own (**NOTE**, if doing the latter and running on a batch file, be sure to include an image as an argument in the batch command).

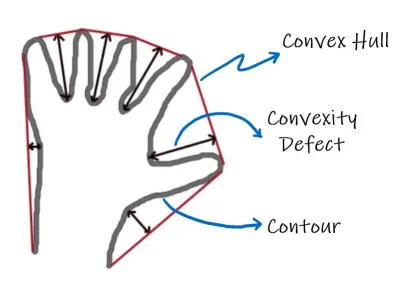
**WHAT WE WERE TRYING TO ACCOMPLISH**

Our primary objectives in developing this interactive game involved accurately identifying the player’s choice of rock, paper, or scissors from the photograph taken. We then aimed to display the outcome of each round, indicating whether the player won or lost against the program. Moreover, to add a layer of complexity and unpredictability, we thought the program should adapt to the player over time: after a few rounds of playing, the program analyzes the player’s previous choices and uses this information to make its own choice, rather than randomly selecting.

**WHAT WE DID ACCOMPLISH**

One of the key achievements of our project was the accurate identification of these hand gestures. It also successfully keeps track of the frequency of each hand gesture made during the game, tallying the corresponding choice the user makes each time they play again. For example, if the ‘rock’ gesture is played frequently by the user, the program will identify this pattern and respond with ‘paper’ (the counter-move for ‘rock’).

We also gathered insight into how the convex hull algorithm is used for gathering information about a hand. Convex hull uses a binary image to create contours of the objects in the image. From these contours, a convex hull creates an outline around the contours and provides us with useful information, (called defects) such as locations where the contour becomes concave and the largest distance between the outline and the concave section. We made use of this information, specifically the number of distances that were within our specified threshold, to determine how many fingers were being held up by the user.



**RESULTS**

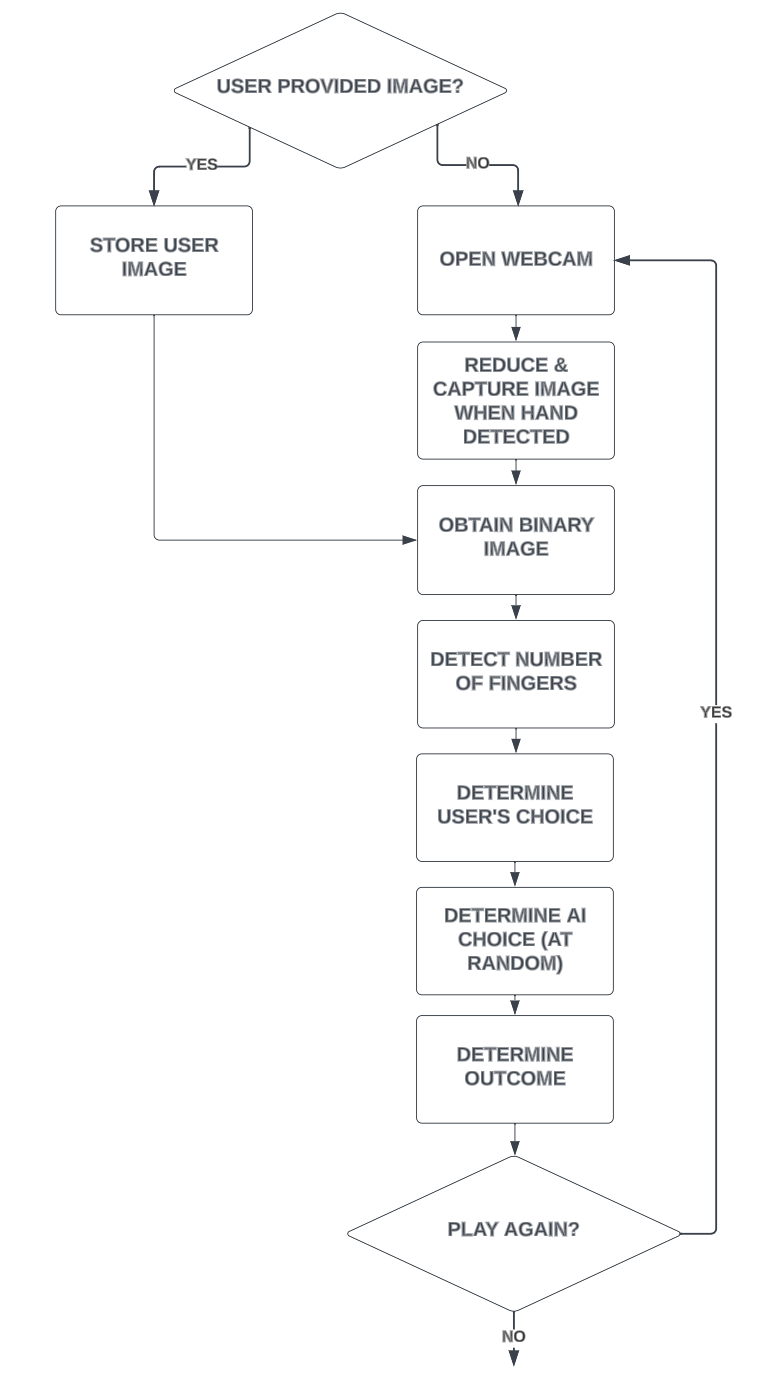
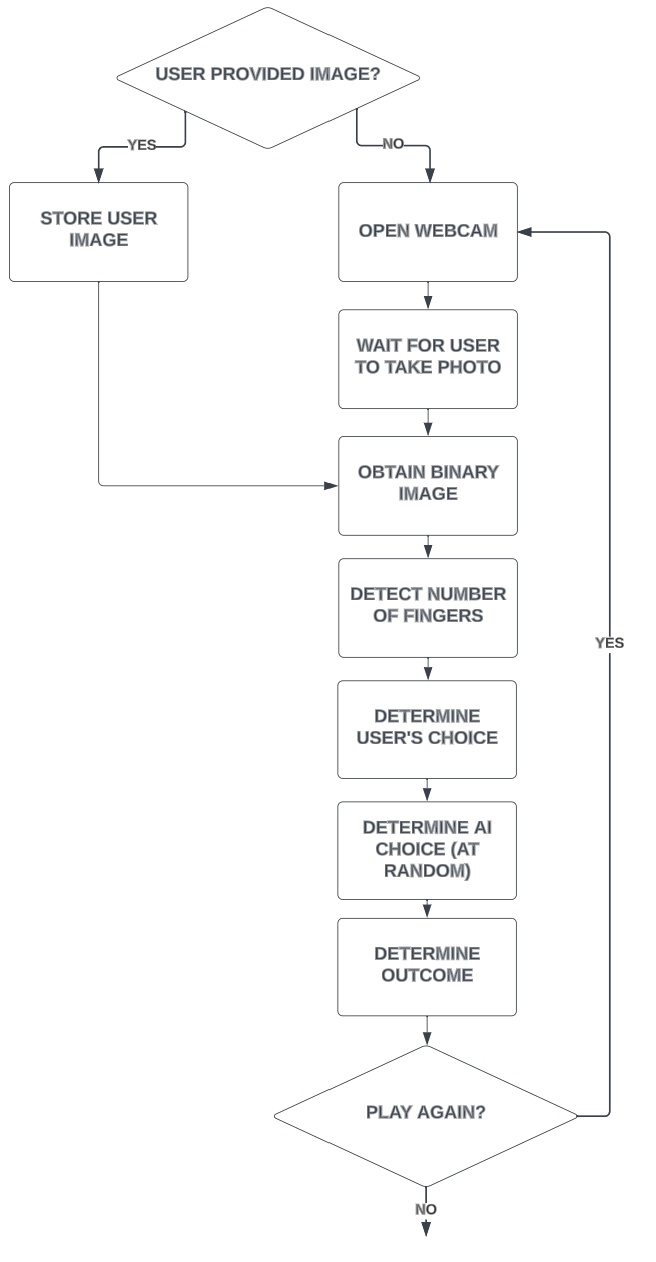
Below is the result of successful detections by the program:

| **IMAGE** | **RESULT** |
| --- | --- |
|  |  |
|  |  |
|  |  |

Below is the result of unsuccessful detections by the program:

| **IMAGE** | **RESULT** |
| --- | --- |
|  |  |
|  |  |

VERSION 1 VERSION 2

****

**LIMITATIONS**

In the development of our interactive Rock, Paper, Scissors game, we have identified several limitations that present opportunities for future research and improvement:

In Version 1 of the system, the image processing algorithm requires a monochromatic background devoid of patterns or visible textures to identify the hand gesture accurately. The quality of the threshold image is significantly influenced by the lighting conditions on the hand, which could potentially impact the game’s results. Moreover, the edge detection technique employed in this version only generates outlines, which may not always yield a comprehensive representation of the hand gesture.

In Version 2 of the system, we incorporated Haar cascades to enhance the gesture recognition capability. However, the accuracy of these cascades is not absolute, occasionally leading to misinterpretation of the player’s choice.

Another constraint is the potential loss of critical hand gesture information when the window is minimized during gameplay. Furthermore, there are instances where the hand detection algorithm fails to recognize the player’s hand, particularly when gestures for paper or scissors are used.

**LESSONS LEARNED**

Throughout the development process of our interactive Rock, Paper, Scissors game, we have gained valuable insights and identified potential avenues for future enhancements.

We learned the importance of leveraging features from OpenCV, such as contours, convex hull, and Haar-Cascade classifiers, in the gesture recognition process. We also realized that the setting of the image, including factors like lighting, background, and textures, plays a crucial role in the accuracy of gesture detection.

One of the key areas for improvement is to increase the accuracy of gesture detection. This could be achieved by refining the image processing algorithm to effectively remove the background and isolate the hand in the image.

Another promising direction for future development is the implementation of a prediction feature. This feature would analyze the player’s previous choices to predict their next move, adding a layer of complexity and strategy to the game.

These lessons learned and identified next steps provide a roadmap for the ongoing refinement of our AI-based interactive game.

**SOURCES**

<https://medium.com/analytics-vidhya/contours-and-convex-hull-in-opencv-python-d7503f6651bc>

<https://pierfrancesco-soffritti.medium.com/handy-hands-detection-with-opencv-ac6e9fb3cec1>

<https://docs.opencv.org/4.x/dd/d49/tutorial_py_contour_features.html>

<https://github.com/Balaje/OpenCV/blob/master/haarcascades/hand.xml>

<https://docs.opencv.org/4.x/d3/dc0/group__imgproc__shape.html#ga8ce13c24081bbc7151e9326f412190f1>

<https://medium.com/analytics-vidhya/contours-and-convex-hull-in-opencv-python-d7503f6651bc>

<https://pierfrancesco-soffritti.medium.com/handy-hands-detection-with-opencv-ac6e9fb3cec1>

<https://docs.opencv.org/4.x/dd/d49/tutorial_py_contour_features.html>