Playing Card Detection and Recognition

Henry Heese
Dept. of Electrical Engineering and Computer Science,
University of Missouri,
Columbia, MO 65211, USA

***Abstract*—In this paper, the faces of playing cards are used as identification targets. A full color image of a playing card is used as input. This image is preprocessed to a 256-grayscale picture, blurred, then converted to a binary image. Contours are found and analyzed to determine the outline of a playing card. Cards are detected using the area and shape of their contour. Using the contour, the image is warped to fit a standard size. Then, the card's identifying information can be isolated and compared against training images. Playing card recognition can assist players with visual impairments, allow for automation of various card games, or monitor gameplay to ensure fairness.**

I. Introduction: Motivation and Background

Playing cards are a common occurrence in everyday life that can be found in entertainment and gambling. Each card having unique patterns and identifiers translates into a nice target for object detection and identification. This paper approaches the application of computer vision in the context of playing cards. Through the use of image processing and object detection techniques, this proposed system attempts to accurately identify and interpret playing cards given diverse conditions. In order to complete this approach in a timely manor, certain criteria were added to limit the scope of the project. This system was created with the standard fifty-two card set of playing cards in mind. This includes the numbered cards two through ten, and the face cards jack, queen, king, and ace. The four standard suits of hearts, diamonds, clubs, and spades are expected. Jokers or wild cards are not considered to be a part of this standard set. The input is expected to have a simple, flat background with consistent lighting. Additionally, cards are expected to have their full face showing without occlusion or overlapping of any kind. The primary dataset used consists of single-card images in which the card is relatively large in comparison to the entire image, and contains variations in rotation and background. This process is intended to be invariant to rotation and perspective as long as the entire card is visible.

II. Image Processing Modules and Pipeline

A. Preprocessing

The first step in this pipeline is preprocessing. Colored input images are converted to grayscale. Then, a gaussian bur is applied to smooth the image and reduce noise. The image is converted again into a binary image using a binary threshold.

B. Card Detection

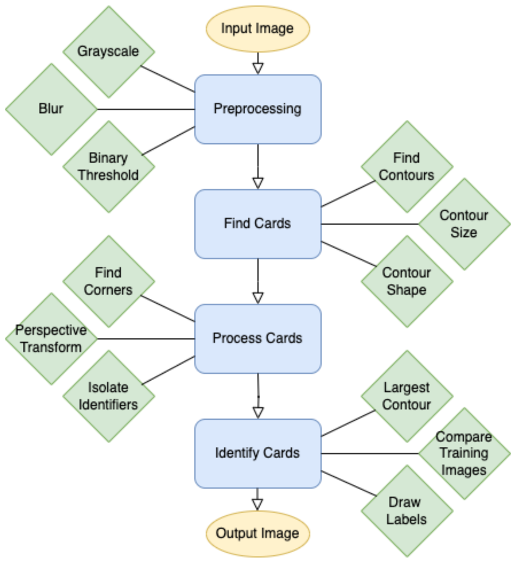
In order to detect playing cards, the binary image is scanned to find all contours within the image. These contours are then sorted and analyzed. Based on the size and shape of the contours, it can be determined with decent accuracy whether or not a contour is that of a playing card. Once a card has been found, its contour information can be utilized in order to process the card further.

C. Card Processing

Using the card perimeter based off of its contour, the bounding box of the card and its corners can be identified. The next step is to perform a perspective transform utilizing the corner information, which warps the card into a standard shape and size. From this standard card, the identifying information in the top corner of the card can be isolated while the rest of the image is cropped out.

D. Card Identification

The final step is the identification of playing cards. Using the cropped sections of the rank and suit information, the largest contour is found which represents each corresponding symbol. With its identifying symbols located, they can be compared against the set of training images in order to determine the best match for rank and suit of each card. With the cards now identified, their labels and borders are drawn onto the initial image as the output for this system. The following image contains a depiction of the overall pipeline sorted into these four sub-processes. Card Identification



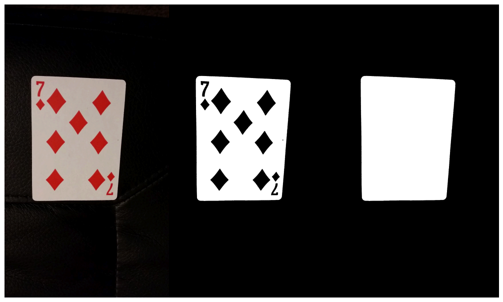
The overall pipeline is similar to what can be found in [5] in which there is a preprocessing and use of contours to detect identifiers. The identifiers are compared against a small template set in both of these pipelines as well. Reference [4] contains a vastly different pipeline to that of this paper. It instead uses deep learning through a neural network in order to identify cards. It first segments the image into many compartments so that each compartment can be analyzed using a YOLO, 'you only look once', method to detect cards with a high detection speed.

III. Approach: Algorithms and Module Implementation Details

This system utilizes a card data structure and and set of training images adapted from their use found in the "RAIN MAN 2.0" blackjack robot [3]. Each card structure stores information related to its contour, dimensions, corners, and its matched rank and suit. It also stores various images used in the process such as the perspective warped image and the cropped identifier sections. The training data contains images corresponding to each suit and rank symbol. The adapted code [3] loads the training images to be compared against the input.

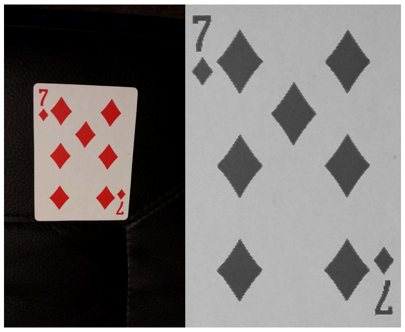
The preprocessing stage consists of three steps. The image is converted to grayscale, the image is blurred using a gaussian blur, then the image is thresholded using a binary threshold. These processes are accomplished using the OpenCV library's convert color, gaussian blur, and threshold functions. Reference [5] utilizes the same three steps in its preprocessing pipeline. The next step in [5] is to analyze the symbols of the cards directly. This allows for more invariance especially against overlapping or occlusion. In this paper, there are additional steps to detect and determine the contour of the entire card. The benefit of this being that the entire card is found, but it does not share the invariance towards overlapping or occluded cards.

To find the cards, first the OpenCV find contours function is applied to the image. This returns all of the contours as well as their hierarchy which can be analyzed to determine which among them are of playing cards. The following graphic shows, from left to right, the original image, the preprocessed binary image, and the image of the identified card contour.

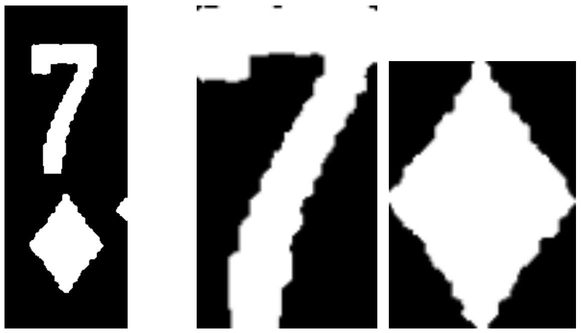


To process the contours, they are sorted from largest to smallest. The area and perimeter are found using OpenCV contour area and arc length functions. They are then analyzed based on four criteria. A contour must be larger than the minimum contour threshold. A contour must be smaller than the maximum contour threshold. A contour must not have any parent contour according to the hierarchy. Finally, they must have four indices indicating that it matches the four-sided shape of a playing card. This last step is accomplished using the OpenCV approximate polygon DP function which utilizes the Douglas-Peucker algorithm to approximate the curves of a polygon. With this information, the system is able to determine if a proposed contour is that of a playing card.

Using the corners identified from the Douglas-Peucker polygon approximation, the next step is to perform a perspective warp on the original image. The perspective warp function [1] transforms a four sided shape into a standardized, flattened image. In this system, it specifically creates a two hundred by three hundred image out of the playing cards for use in further processing. This is accomplished by identifying each corner as top or bottom and left or right. This process is invariant to card rotation. It can determine if a card is vertical, horizontal, or diagonally tilted by comparing the width and height of the card's bounding box. The following image shows a section of the original image image on the left as well as a grayscale, perspective warped version of the image on the right.



Converting the card into a standardized format allows for easier recognition of the rank and suit. Since the standard form will always have the identifying information in the same place, correctly detecting a card should guarantee that the identifying information can be located as well. Using preset size values, the corner of the card is cropped off of the rest of the image. It is also split in two so that there is an image of the card's rank and a separate image of the card's suit. Below is an example of the cropped card corner followed by the separated rank and suit identifiers from the card image.



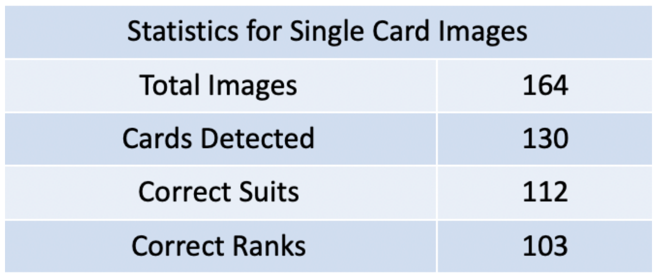
With the identifiable components isolated, they can now be compared against the training images. This is performed by taking the absolute difference of the test image against each training image for both the rank and suit categories. Specifically, this involves subtracting the training image from the test image. The result is then converted to its absolute value. Then, the summation of each pixel in the absolute difference gives the difference value for each testing image. The lowest difference is kept for each as long as it is within the preset maximum bound for the difference. If none of the training images generate a difference within the preset threshold, the unidentifiable rank or suit will be marked as unknown.

The last step adapts a function to draw the results onto the card [3]. This displays the card's rank and suit at the center of the card. This is different to reference [5] in which each symbol is identified separately and is then drawn next to the symbol instead of the center of the card. A benefit of this system is that each card is only processed once whereas reference [5] processes each symbol individually. This results in redundant processing for the duplicate symbols on each card. It also requires further processing to remove duplicates when determining the full set of cards displayed.

IV. Experimental Results and Discussion

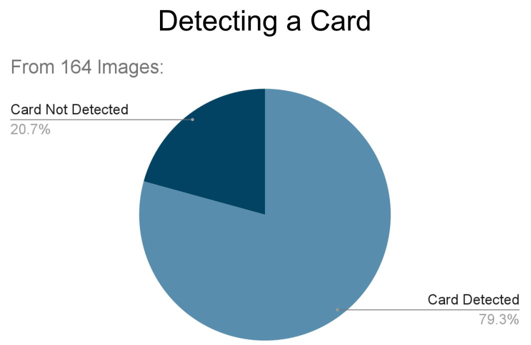
The majority of the testing of this system was conducted using a database of images in which a singular, face-up playing cards is located at the center of the image [2]. This database allows for the system to be tested in a simple environment within the guidelines of the given scope. The set contains occurrences of every card in a standard set portrayed on top of a variety of backgrounds with varying rotations and lighting. The total set contains 216 images, however some of them have been discarded due to containing a joker card or too complex of a background. Each of these cases falls outside the scope of this paper. This leaves 164 images which were used for testing. Each of these images was run through the system to attempt to identify the rank and suit of the card within the image. The dataset [2] includes the rank and suit of each depicted card within the name of the file. The rank and suit are snipped from the file name to be compared with the result of the system's detection and recognition procedure. The resulting statistics are discussed below and displayed in the following graphics.

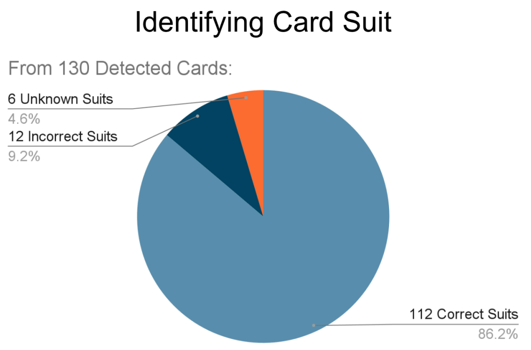
Out of the 164 images that were processed, 130 of them returned a detected playing card. This accounts for 79.3% accuracy in detecting cards from the single-card images and a 20.7% failure rate in card detection. The system was able to correctly identify 112 of the cards' suits. Of the 130 detected cards, 86.2% had their suits correctly identified. This process also resulted in 6 cards having unknown suits and 12 cards having incorrect suits. For the card ranks, 112 were identified correctly resulting which is 79.2% of the cards that were detected. The rank identification also had 2 unknown ranks and 25 incorrect ranks.

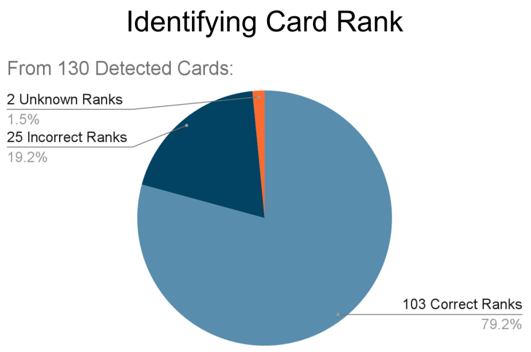


Statistics for Single Card Images	
Total Images	164
Cards Detected	130
Correct Suits	112
Correct Ranks	103

The initial tests of the system resulted in worse results, but by modifying parameters the statistics have been improved to what they are now. The maximum and minimum area thresholds for detecting card contours were adjusted until no false positives were detected. Previous iterations falsely detected cards in the background, but the current program did not falsely detect a card where a card was not present. Extending these parameters could potentially allow for a higher percentage of cards to be detected at the cost of including false positives. Reducing false positives was determined to be more significant.







Modifications to the maximum matching difference also affected the results of the system's output. Increasing the threshold for rank and suit identification resulted in more correctly identified identifiers up until a certain point. After that point, extending the range resulted in less unknown identifiers but equivalently more incorrect identifiers. Common mismatches include a '3' being mistaken for an '8' or a '9' being mistaken for the '0' of a ten. These common mismatches reflect a similarity of the symbols. The suit identification resulted in slightly higher accuracy than the rank identification. This is likely influenced by having significantly less suits than ranks for the system to compare against. Having separate rank and suit training sets does not seem to have a significant advantage. The identifiers of one category are significantly different enough to the other that a rank test image and a suit training image are not likely to be mistakenly matched [5].

V. Conclusion and Future Work

Future work regarding this paper could work on extending the scope and allow for more variants of images to be used. A significant limit to the scope is that it expects cards to not be overlapping or occluded. Accounting for this may require a restructuring of the pipeline to be more in line with reference [5] such that detecting the entire card is not necessary and instead each symbol is detected and identified individually.

This paper's matching could also be improved by implementing additional methods to assist in recognition of a card's identifying information. A potential improvement to this project's matching could be to implement connected component labeling in order to count the number of symbols at the center of each card. This method should be able to assist in identification of the ace through ten numbered cards since they each have their rank represented by the number of symbols within the center of the card.

This project could also potentially be adapted to apply to a more specific circumstance. For example, the output could be examined in the context of a poker hand with insights as to how a player should act regarding the current hand.

References

1. A. Rosebrock, "4 Point OpenCV getPerspective Transform Example," *PyImageSearch*, Aug. 25, 2014. https://pyimagesearch.com/2014/08/25/4-point-opencv-getperspective-transform-example/
2. B. B. Lohray, "Playing Cards Detection," *GitHub*, Aug. 21, 2023. https://github.com/lordloh/playing-cards/tree/master.
3. E. Juras, "RAIN MAN 2.0 - Blackjack Robot," *hackaday.io*, Feb. 20, 2018. https://hackaday.io/project/27639-rain-man-20-blackjack-robot
4. Q. Chen, E. Rigall, X. Wang, H. Fan and J. Dong, "Poker Watcher: Playing Card Detection Based on EfficientDet and Sandglass Block," 2020 11th International Conference on Awareness Science and Technology (iCAST), Qingdao, China, 2020, pp. 1-6, doi: 10.1109/iCAST51195.2020.9319468.
5. X. Hu, T. Yu, K. Wan and J. Yuan, "Poker card recognition with computer vision methods," 2021 IEEE International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, 2021, pp. 11-15, doi: 10.1109/ICETCI53161.2021.9563607.