Machine Vision in Casino Game Monitoring

João Paulo Maurício Pimentel

Abstract—To prevent game losses due to mistakes or irregular strategies, or for statistical data acquisition, the casino game industry is interested in automatic monitoring of the games, recording their state for posterior analysis.

A method for detection and recognition of playing cards taking into account partial occlusion and rotation is proposed. This is carried out in two phases: a scene is analyzed to detect rectangles with an original method, fitting the card size, and then each detected rectangle is classified according to the figure present on the corner of the card. The cards are all assumed of a known size and scale, and the image to be analyzed has no perspective or optical deformations, although a calibration phase to take this into account has been developed and tested.

Three methods for the card recognition stage are studied and compared, evaluating their performance in terms of robustness to brightness and contrast changes and computation time. A template matching method was shown to work successfully and time efficiently, although showing low robustness to lighting changes, whereas an edge based probabilistic rigid model showed robustness to such changes at the expense of a longer computation time. A probabilistic deformable model presented low success and a very long computation time.

Index Terms—Playing Card Recognition, Rectangle Detection, Object Recognition, Template Matching, Edge Based Probabilistic Model.

I. INTRODUCTION

The current computer vision technology has brought a rise in automatic video-surveillance solutions. This solutions are normally implemented for security reasons, but are also used in statistical applications, such as people or vehicle counting systems. In this sense, the casino game industry is interested in automatic monitoring of the games, either for security (making sure the rules are correctly followed) or for statistical data acquisition.

This project aims at developing of a playing card detection and recognition robust enough to perform well with any rotation of the cards, as well as cards partially occluded by others.

A. Problem Formulation

The application was developed for Anglo-American playing cards, as can be seen on Figure 1. In this project, the index of the card is defines as the image of the right upper corner of the cards, made of a rank on the top, which can take one of 13 values (A,2,...,10,J,Q,K), and a suit on the bottom, taking one of 4 values (Clubs, Spades, Hearts, Diamonds).

The problem is described as follows: Any number of cards are laid on a dark surface in any rotation. Any number of cards can be on top of another, as long as at least some of the outer edge of every card is visible, and the index e fully visible.

![Figure 1: Type of card and its components used in this project.](image)

II. STATE OF THE ART

There has not been much scientific work on card detection and recognition. The two main works reviewed [1], [2] approach the problem in a similar way, first by detecting the card and then classifying it. This methods are based on contour extraction, in a way such as the detection cannot be robust to partial occlusions. In [1] the detection is performed only for an isolated card, detecting the rectangle fitting a bounding box and then estimating the rotation of the card. The card is classified using a template matching method for the rank and suit.

In [2] the detection is based on closed contours of the image. A contour with 4 vertices and a given area is considered a card contour. Any contour with another given area and aspect ratio is considered a suit block (the suit symbols in the interior of the value cards - cards with rank A, 2 to 10). The classification is made with template matching of the contours, counting the number of suit block in the interior, as this number correspond to the rank of the value card. The face cards (J,Q and K) are classified using a template matching method for the rank and suit.

III. CARD DETECTION

The first step of the proposed approach is the card detection. This is carried by a voting scheme, based on the Generalized Hough Transform [4]. A card is modeled as a rectangle defined by the center localization $c = (c_x, c_y)$ and orientation $\phi$. Since the cards are rotationally symmetric for rotations of $180^\circ$, $\phi$ is defined in $[0, \pi]$. The 0 degree orientation is considered to be the orientation with the longer edge of the card in the horizontal. We define the pose of a card as $\theta(c, \phi) \in \Theta$, the space of admissible poses. As the card size and scale are known a priori and having a rectangle described using only these three parameters, any non-rigid transformation on the image is not allowed.
**A. Hough Transform for Rectangles**

The voting accumulator for the Hough Transform is defined in the three dimensional space $\Theta$. The outer contours of the cards in the image are extracted, as well as the gradient direction in such points. Knowing the gradient in a contour point is directed to the inside of the card and perpendicular to the contour, each contour point can only belong to rectangles in just two orientations: the same as the gradient orientation, assuming the contour point belongs to the shorter edge of the card, or the gradient orientation plus $90^\circ$, assuming the edge point belongs to one of the longer edges of the card. For each of these orientations, there is a finite number of possible rectangles passing in said contour point. The number of possible rectangles is proportional to the size of the edge we are assuming the contour point belongs to, given that the pose space is discrete.

Let $M(x)$ be the contour map, having $M(x) = 1$ when there is a contour in $x = (x,y)$ and let $\Phi(x)$ be the gradient direction at $x$. The card size is defined as $2R \times 2r$, with $2R$ corresponding to the size of the larger edge of the card, and $2r$ the shorter one. We define a voting accumulator $H$, indexed by the elements of the pose space $\Theta$. Each contour point votes on the possible respective rectangles by incrementing the values of $H(\theta_i)$, with $\theta_i \in \Theta$ being the poses of said rectangles. The voting procedure is described as follows:

For each point $(x,y)$ such that $M(x,y) = 1$, select the rectangle centers of the candidate rectangles to be voted. These are given by

$$c^j_R = x + rv + jw, \quad j = -R, ..., R \quad (1)$$

$$c^k_r = x + Rv + kw, \quad k = -r, ..., r \quad (2)$$

with $v$ and $w$ being unit norm vectors with direction equal to $\Phi(x)$ and $\Phi(x) + \frac{\pi}{2}$, respectively. The centers $c^j_R$ correspond to the hypothesis of $x$ belonging to one of the longer edges of the card, and $c^k_r$ correspond to the centers in the hypothesis of $x$ belonging to one of the shorter edges. These centers belong to a line segment parallel to the hypothesized edge, passing by the center of the card, with a length equal to the size of the non hypothesized edge (the points $c^j_R$ are at a parallel distance of $r$ from $x$, and $c^k_r$ are at a parallel distance of $R$). A visual representation of this points, for a contour point $x$, is present in Figure 2.

Knowing the two only possible orientations of the candidate rectangles, as described above, the poses $\theta_i$ to increment the accumulator $H$ will be given by the set $\Theta_c \subset \Theta$:

$$\Theta_c = \{ \theta^j = (c^j_R, \Phi(x) + \frac{\pi}{2}, j = -R, ..., R) \} \cup \{ \theta^k = (c^k_r, \Phi(x)), k = -r, ..., r \}. \quad (3)$$

which corresponds to the union of the poses of the two sets of centers, each one with its orientation.

To compensate for possible sizing errors, for example, an error of a single or little more pixels in the card dimensions, the actual incrementing of the accumulator is not done only on the poses described above, but also in a small neighborhood around them. More precisely, the centers of the candidate rectangles used are the same as the above plus an additional set of centers, located at a parallel distance of 1 pixel from the original centers, from both sides. The orientations of the additional poses are the same as in the original poses. This can be viewed as a rectangular blur of the accumulator in just the localization dimensions.

**B. Rectangle detection**

After looping for every contour point the filling of the accumulator is over. Every point in the accumulator represents a rectangle pose, and its value correspond to the number of votes of that pose. The detected rectangles are represented by the local maxima of the accumulator. These are detected efficiently by looping only through the accumulator points with at least 1 vote, selecting as rectangles those poses which have a value above some defined threshold, and a value higher than or equal to the values of the 26 neighbors.

Using a voting scheme to detect rectangles, any card having just a small part of its contour visible can be detected, successfully solving the problem for partial occlusions. However, this also results in a large number of false positives. The three main cases of this are: several detected rectangles near the correct one; rectangles detected having part of the contour in the exterior of the card image; and too many detected rectangles when the cards are occluded but aligned with one another. To attempt to solve this kind of problems, we loop through the rectangle candidates, ordered by the number of votes (their ”strength”), and each one is analyzed sequentially.

To solve for the excess of detected rectangles near the correct one, a non maxima suppression is performed. Each rectangle which is in a small neighborhood (in terms of the pose space) of a stronger rectangle is rejected.

If a rectangle passed through the previous test, the image of its interior is analyzed using the contour map as the image. If there are contour points inside the rectangle this cannot be a real card, because only the outer contours of the cards are present in the image, and as such it is rejected. Account
for small deviations on the perfect position of the detected rectangles is taken.

At last, when the cards are occluded by alignment either by the short sides or the longer ones, the image of the contours does not give enough information about the number of cards present. To minimize the number of detected rectangles in this situation, the image of a small neighborhood of the corners of the rectangles are analyzed, to determine if the number of contour points matches an expected number of contour points in a real card corner. This very simplistic corner evaluation is not very significant, as it was not intended to spend too much computation time in the rectangle detection.

The detection phase is tuned as to bring no false negatives, at the expense of a possibly high number of false positives. Since any rectangle that passed this filtering can be rejected in the card classification phase if no index is found on its top right corner, these false positives do not pose a big problem for the method.

IV. CARD CLASSIFICATION

The second phase of the proposed method is the card classification. After the detection, and having a correct pose for at least the cards present in the image, their index can be easily extracted, rotating the image in the detected rectangle and retrieving the image of the top right corner. Thus, the index images are assumed well segmented. Classification was done using various methods for comparison. The rank of the card is first classified, and if considered a valid card (likelihood or other matching criteria above threshold) the suit is also classified. Otherwise, the card is rejected.

A. Probabilistic Rigid Method

A method based on [5] was implemented. It considers binary oriented edge features, in 8 orientations at increments of 45°, computed at each point $x$ of a grid $L$. Let $X = \{X_e(x)|x \in L, e = 1, ..., 8\}$ be the set of features with orientation $e$. Having $X_e(x) = 1$ means a feature with orientation $e$ is present at $x$. After the feature extraction, the maps of the detected features are dilated with a $3 \times 3$ kernel. This dilation seeks to bring robustness to small deformations and enhance the classification.

Even if the objects to be classified (rank or suit images) are correctly segmented, this does not imply a constant position on the grid. Given this, and that the model is rigid, the pose $\theta$ of the object is defined as a 2D vector $r$. We define a probability map $p_c(x)$, defined on the grid $L$. Given the object is present at a location $r$, we assume the edges present are conditionally independent with marginal probabilities given by

$$P(X_e(x) = 1|\theta) = p_e(x; \theta) = p_e(x - r). \quad (4)$$

Being this a rigid model, the probability map is shifted by $r$. To model the presence of edges outside the object (the background), we define the object support $S_e(\theta) = S_e(r)$ for each edge direction, and assume a background model outside of $S_e(r)$, having the edges independent as well, but with a homogeneous marginal probability equal to $p_{e,\text{bgd}}$. The support is defined as the set of points where the marginal probability is greater than a threshold $p_{\text{min}}$. With this defined, the likelihood of the edge data is given by

$$P(X|r) = \prod_{e} \prod_{x \in S_e(r)} [p_e(x; r)]^{X_e(x)} [1 - p_e(x; r)]^{1 - X_e(x)} \times \prod_{e} \prod_{x \notin S_e(r)} [p_{\text{bgd}} X_e(x)]^{1 - p_{\text{bgd}}} \cdot X_e(x), \quad (5)$$

corresponding to a binomial distribution on the feature presence at point $x$.

Defining $P_c(X|r) \doteq P(X|c, r)$ as the likelihood of the edge data given the object belongs to the class $c \in C$, the space of the different classes, the class of the object being analyzed is given by the maximization

$$\hat{Y} = \arg\max_c \max_r P_c(X|r). \quad (6)$$

For each class, the probability maps for that class are shifted by every value of $r$. The classification is done choosing the class with maximum likelihood, if that value is above threshold. If not, the detected card is rejected.

To train the model, i.e. the probability maps, the edge features are extracted on a set of training images. The probability maps of the model are given by the average of the edge feature maps of the training images, for each edge direction $e$.

B. Deformable Probabilistic Model

A deformable model proposed in [5] was also implemented. Having the detection and recognition problem rigidly posed, with non rigid deformations not considered, there is not much motivation for this kind of model, thus it was not extensively explored in this work. It is similar to the model described in the previous section. The same binary edge features are used, and a probability map $p_e(x)$ on feature presence is defined in the same way.

To model the deformations of the object, the pose cannot be described just as a location vector as in the rigid case. We define $n$ reference points $(y_i)_{i=1,\ldots,n}$. The deformation is described by a set of $n$ 2D shifts $v$ and a location $r$ as well, defining the pose as $\theta = (r, v) = (r, v_1, \ldots, v_n)$. Each $y_i$ is mapped to $z_i = r + y_i + v_i$, with $r + v_i$ representing a rigid shift of the model. This way, the deformation is described by the shifts of the reference points. If all $v_i$ are equal, the pose can be described as a location $r$, hence having no deformation present. To consider different shifts of the reference points, different models are shifted accordingly, and combined with an averaging operation, performed by parts.

We define a part $Q_t$ associated to the reference point $y_i$ as a submap of $p_e$ in a square neighborhood $W$ around $y_i$:

$$Q_t \doteq p_e(y_i + s), \quad s \in W, \quad e = 1, \ldots, E. \quad (7)$$

The map $p_e(x; \theta)$ can no longer be given by a shift of $p_e(x)$ as in the rigid case. The parts are combined by a Patchwork of Parts (POP) operation. Let $N(x) = \{i : x \in z_i + W\}$ be the set of windows $W$ covering the point $x$. The POP operation is used to compute the map $p_e(x; \theta)$ as follows:

$$p_e(x; \theta) = \begin{cases} \frac{1}{|N(x)|} \sum_{i \in N(x)} p_e(x - z_i + y_i) & \text{if } N(x) \neq \emptyset \\ 0 & \text{if } N(x) = \emptyset \end{cases} \quad (8)$$
The map \( p_c(x; \theta) \) is given by the combination of the different parts, computing the average in the regions covered by more than one part. Having the probability maps \( p_c(x; \theta) \) defined, defining the object support and assuming the edges are conditionally independent as in the rigid case, the likelihood of the edge data is given in the same way as (5):

\[
P(X|\theta) = \prod_x \prod_{e \in S_e} [p_e(x; \theta)]^{X_e(x)}[1 - p_e(x; \theta)]^{1 - X_e(x)} \times \prod_{e \notin S_e} \prod_r [p_{bgd}(x; \theta)]^{X_{bgd}(x)}[1 - p_{bgd}(x; \theta)]^{1 - X_{bgd}(x)}. \tag{9}
\]

The classification is done choosing the class with maximum likelihood as before:

\[
\hat{Y} = \arg \max_r \max_v P_e(X|\theta = (r, v)). \tag{10}
\]

However, the maximization over \( v \) is difficult, since there can be a very large number of possible combinations of the vectors \( v_1, \ldots, v_n \). To choose the shifts \( v_i \) that give the maximum likelihood for a given class and location \( r \), one of the methods proposed in [5] is used, called Independent Maximization:

Initialize \( v_i^{(0)} = (0, 0), \quad t = 1, \ldots, n \). Choose a small square neighborhood \( V \) around the origin. At iteration \( t \) loop through the different reference points \( y_i \). For each vector \( v \in v_i^{(t-1)} + V \) the part \( q_i \) is placed at \( y_i + v \) and its likelihood is computed using (9) only on the parts of the part, ignoring all the others. The value of \( v \) that corresponds to the maximum likelihood is given to \( v_i^{(t)} \). After all \( v_i^{(t)} \) are updated, more iterations can be made. In other words, the shift associated to each reference point is the one that gives maximum likelihood of the part associated to it.

After the best shifts are \( v_i \) are selected, this maximization is repeated for every value of \( r \) and for each class, and the class chosen is the one with maximum likelihood, being rejected if this value is below threshold.

The model training is analogous to the rigid case, except that it is made for each part. Defining the reference points \( y_i \) in a regular grid, and after the feature extraction of the training images is performed, the average of the feature map areas corresponding to each part is computed, having each part as the average of the parts on the training images.

### C. Template Matching

Having a fixed scale imposed on the card images and having the orientation of a detected card known, a natural choice for a classification method is the template matching. Each rank or suit of the detected card is compared with a template from each class, using a cross-correlation criteria. As before, the class chosen for the rank and suit is the class with maximum cross-correlation, if it is above threshold.

The templates used are constructed using the average of the training images.

### V. DETECTION AND RECOGNITION IN PERSPECTIVE

To demonstrate that the proposed method’s efficacy is not entirely dependent on the camera position being perpendicular the card plane, it was tested taking the images with different camera positions and angles. An image taken this way presents some perspective deformation, which was corrected as to proceed to the card detection phase.

To correct the perspective deformed images, the pinhole camera model was used. By this model, the image is formed by projecting 3D points in the physical world into the image plane, according to a perspective transformation:

\[
sq = M \begin{bmatrix} R|t \end{bmatrix} Q, \tag{11}
\]

or

\[
s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{12}
\]

where \((X, Y, Z)\) are the coordinates of a 3D point in the physical world and \((x, y)\) are the coordinates of the projection point in the image plane in pixels. \(M\) is the matrix of intrinsic parameters of the camera, which are the focal lengths \(f_x, f_y\) and the principal point \((c_x, c_y)\), usually at the image center. This parameters do not depend on the camera position. The joint rotation-translation matrix \([R|t]\) is called the matrix of extrinsic parameters, and describes the camera position, by translating a point \((X, Y, Z)\) to some coordinate system.

Given the cards to be recognized are all in the same plane, there can be a 3D coordinate system such that the card plane corresponds to \(Z = 0\). Using this value, the third row of the matrix \(R\) can be rejected, ending with a projection described by

\[
s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{13}
\]

where \(H\) is a \(3 \times 3\) matrix called homography matrix. Having the projection described this way, we don’t need to explicitly estimate the camera intrinsic and extrinsic parameters, but the whole matrix \(H\) can be estimated using at least 4 correspondences of points in the image plane and in the card plane.

The calibration used in this work follows this. A calibration image is acquired, with a rectangular piece of paper of known size. A corner detection algorithm, based on the minimum eigen values of a covariance of derivatives matrix, is executed on the image acquired to detect the 4 corners. Knowing the size of the paper, we can correspond its corners to 4 known locations on the image, respecting the aspect ratio of the paper. Calibrating this way we get an image with correct appearance, solving the perspective deformation problem.

### VI. RESULTS

The results for each phase of the proposed method, the detection phase and the recognition one, will be shown. Having used three different methods for the recognition phase, their results will also be shown for comparison. The results of the calibration procedure are also shown.
A. Card Detection

The detection phase, when the card size is modeled carefully, delivers no false negatives. It takes a considerable amount of time (1 second for an image with about 13 cards). Partial occlusions pose no problems, except when there are several cards aligned vertically or horizontally. In these latter cases, as was described above, the outer contours do not give enough information about the number of cards present in the middle. These cards cannot be guaranteed to be detected and recognized. The number of false positives detected depends heavily on the cards disposition in the plane. When they are disposed in a way such that no cards but the real ones can fit the interior of the contours, the number of false negatives can be and typically is null. In the reverse situation, a large number of false positives can appear. The Figure 3 show an example of this, with the correctly detected and classified cards delimited in red, and the false detected positives in green after rejection in the classification phase.

When at least a whole corner of a card is visible, the detection faces no problems.

B. Classification

The classification phase is the most sensitive one. Having a possibly large number of false detection positives, the classifier should be able to correctly classify just the real detected rectangles, and reject any other. Both the rigid probabilistic model and the template matching perform this task. Given ideal illumination conditions (the same as in the training images) both these methods deliver 100% correct recognition rates.

The illumination can be simply modeled as brightness and contrast. A set of images was analyzed with several values of brightness and contrast by both rigid methods (probabilistic model and template matching). The rigid probabilistic model, being based on edge features, is relatively invariant to illumination conditions. The success of the template matching can depend heavily on these conditions. This is shown in Figure 4, where the recognition rates with the probabilistic method are shown. The probabilistic rigid method behaves well for large variations of brightness and contrast, whereas the template matching only works in small variations on the training images. The variation values are considered as 100 being the most positive variation possible, and -100 the most negative one. The probabilistic method is, however, a lot more computationally expensive when compared to the template matching, resulting in a performance as much as 200 times slower. The template matching can take as little as 0.02 seconds in an image with 13 cards.

A great part of the classification errors with the template matching come from the suit recognition. The suit symbols are not very different from one another in terms of shape, when compared to the different ranks, and the red suits appear less dark than the black ones when the image is in gray scale. These facts make the method incorrectly classify a black suit as a red one when, for example, the image has a brightness increase over the ideal illumination, making a black suit symbol have the same gray scale value as a red suit.

The deformable model works extremely slow. Even though it correctly classifies a card rank when its image is deformed, e.g. by an affine transform, the model does not work successfully in normal conditions, failing considerably. This can be explained by the fact that the deformation estimate can deform the object to the point of being close enough to a wrong class, since no deformation is expected, resulting in a misclassification.

C. Classification with Perspective Correction

The perspective correction was tested using the original image in two different resolutions, to assess how the corrected image appears as it is expected (with no deformation and correct reconstruction) according to the source resolution. The quality of this calibration proved to be sensible to both the source image resolution and the degree of perspective, i.e., how much is the perspective deformation, which depends on the angle of the camera with the card plane. Figure 5 shows an example of a heavily perspective deformed image of a card, and the corner of the card in the corrected image for two different source resolutions. A low source resolution results in a much worse reconstruction of the image. This can be because of interpolation errors, since with a lower resolution the pixel values have to be computed from longer interpolations, and
also because the corner detection is not so accurate, when compared to a higher resolution.

The image correction problems reflect on the classification performance. The card detection works well with the corrected images, since the correction, even when not very accurate, does not change the card edges enough to affect it. The images of the rank and suit symbols can, however, become too degraded. When a high enough source resolution is used, both the rigid methods perform with 100% recognition rate. When the source resolution is low, as in figure 5(a), the probabilistic rigid model presents a lower recognition rate, as low as 79.5%. The template matching however presents perfect results in the same images. This can happen because the degradation of the corrected images affect the feature extraction, failing to detect edges where they were expected.

Even if there is a small deformation of the symbols in these cases, the deformable model fails to correctly classify the cards. The deformable model is also dependent of a correct edge feature extraction.

VII. Conclusion

A method for detection and recognition of playing cards was proposed. An original method, based on the Hough transform, for the detection of cards as rectangles was developed. Being a voting scheme, the detection performs well in cases of partial occlusion. For the classification, made independently from the detection, three methods of object recognition were compared: template matching and two probabilistic methods, one rigid and the other deformable. With controlled illumination conditions, the best method for this application is the template matching, for the efficacy and computational efficiency. The probabilistic rigid model works well in various illumination conditions, but is slower and presents some problems when not enough definition in a perspective corrected image is available. The probabilistic deformable model is too slow and presents low efficacy for this rigid application, not being a good method to consider.

As is natural in any work with limited time, several other paths could have been explored or further developed. The proposed method simulates the human vision in card recognition only for a simple and controlled case. In fact the human being is able to identify the cards even without observing their contours, searching and finding a rank symbol with a suit symbol below. He can also identify the cards even without the index image, since the card interior contains all the information needed about it. This way, some of the paths that could have been explored are presented.

Rectangle Detection. The detection presented assumes the cards have no deformation and a fixed scale and dimension. A more robust detection could find this cards even under perspective. The card detection itself and the consequent perspective detection could be used to calibrate the image, as to obtain the the index images with no deformation and in the right scale. However, even if the method would become more robust, the classification would still depend on the contour presence.

Detection of the card indexes. One way to integrate the detection and classification in just one phase is to search the image for card indexes (ranks and suit symbols below). A detection algorithm based on robust features, such as SIFT[3] or SURF[6], could be used. This way, the detection would not depend on the visibility of the contours, and to a certain extent would be robust to deformations such as perspective. The classification would be closely related to the detection, identifying each candidate region or blob as belonging to a card, being classified or rejected.

Classification of the inner image. The image in the interior of every card contain all the information needed to classify it. In the rank cards (A, 2,...,10) the rank is given by the number of suit symbols in their interior. These symbols have all the same size, except for the Ace of Spades, which typically has a different suit symbol referring to the card manufacturer. The classification can be made by counting the symbols, in case they are all visible, or by their disposition, which varies according to the rank and can be used in cases of partial occlusions (with limitations). The face cards (J, Q, K) have an image in the interior. Even though it’s not easy for a human observer to identify a face card with just this image, every face card has a different one. This can be detected and classified with a SIFT or SURF descriptor, even with some occlusion.

Every method integration. The human being uses every one of this processes to identify playing cards. A robust method could work the same way, starting from a simpler method and climbing to the more complex ones when needed.

References