PLAYING CARD DETECTION

19ucc023 – Mohit Akhouri

19ucs064 – Tapomay Singh Khatri

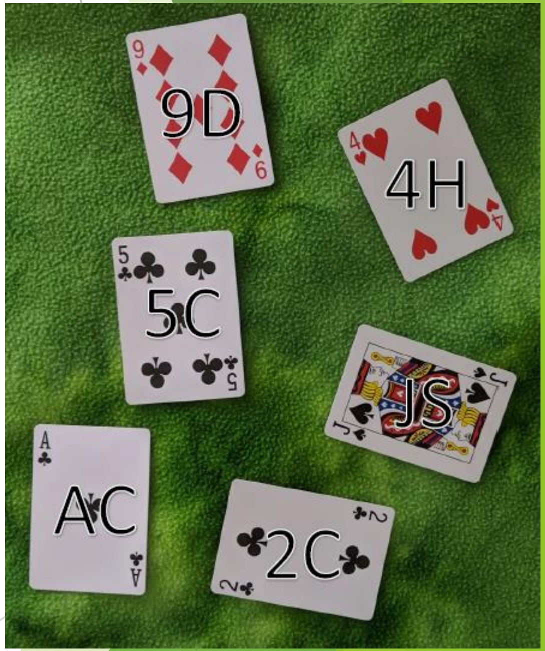
19ucs005 – Prasanna Mishra

19ucs024 – Hardik Satish Bhati

19ucc105 – Rhythm Bajaj

GOAL OF THE PROJECT

- The goal of the project is to correctly identify and locate any of the cards in a standard 52 card deck. The program will work in the following conditions :
  - Images of a single card on a variety of backgrounds.
  - Images where the card(s) are not centered and are rotated.
  - Images containing multiple non-overlapping cards.
- Technical Approach which we will follow : The algorithms which we will be using will be using these 3 steps to accomplish the task. The 3 steps are :
  - Card Detection : The area a card occupies within the image will be identified. Techniques used are gray level thresholding and followed by edge detection.
  - Card Rectification : The region from detection is rotated so that edges lie along x and y axes. Can be done using hough transform. Image can be resized to enable template matching.
  - Card Identification : The card's suit and rank is identified. Some possible approaches used are template matching and general optical character recognition techniques.



DATA COLLECTION

- We will collect two sets of data for this project. The one set is the training images used to develop the templates. The other set is the test set used to test the implementation of the algorithm. The first set is referred to as **training/ template images** and the second set is referred to as **target images**.

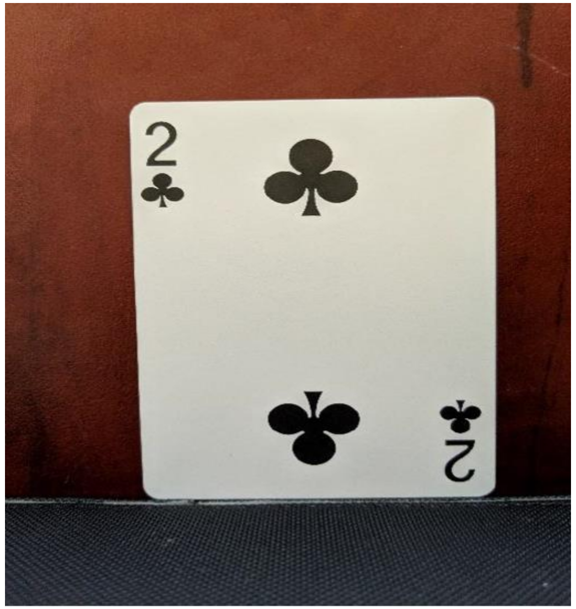


Figure 1: Example Training Image

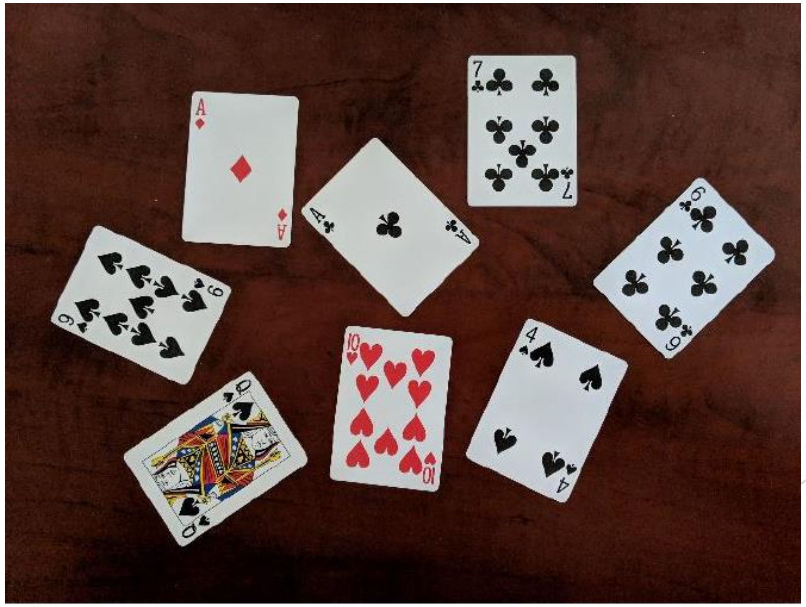


Figure 2: Example test image

BACKGROUND REMOVAL

- Using OTSU's Method : Here , we can use Otsu's method to perform automatic image thresholding. The algorithm returns a single intensity threshold that separate pixels into two classes – foreground and background. This method searches for the threshold intensity I_t which maximises the between class variance σ_b^2 .

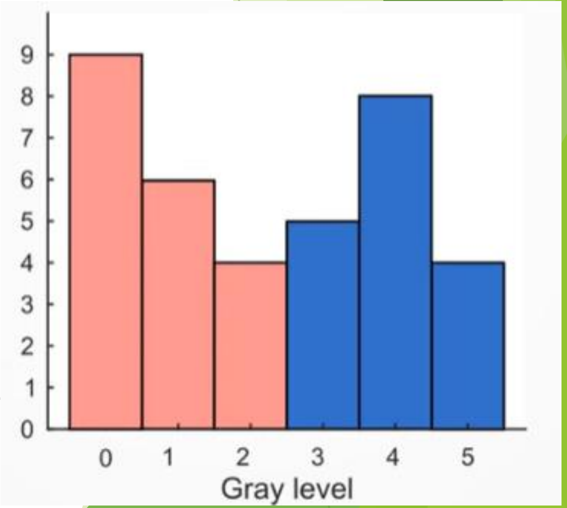
$$\sigma_b^2 = W_b W_f \ (\mu_b - \mu_f)^2$$

$W_{b,f}$ = Number of pixels in background (foreground) / Total number of pixels

$\mu_{b,f}$ = Mean Intensity of background (foreground)

- Limitation of Otsu's method – Works well if the histogram has a **bimodal distribution**.

Otsu's Method

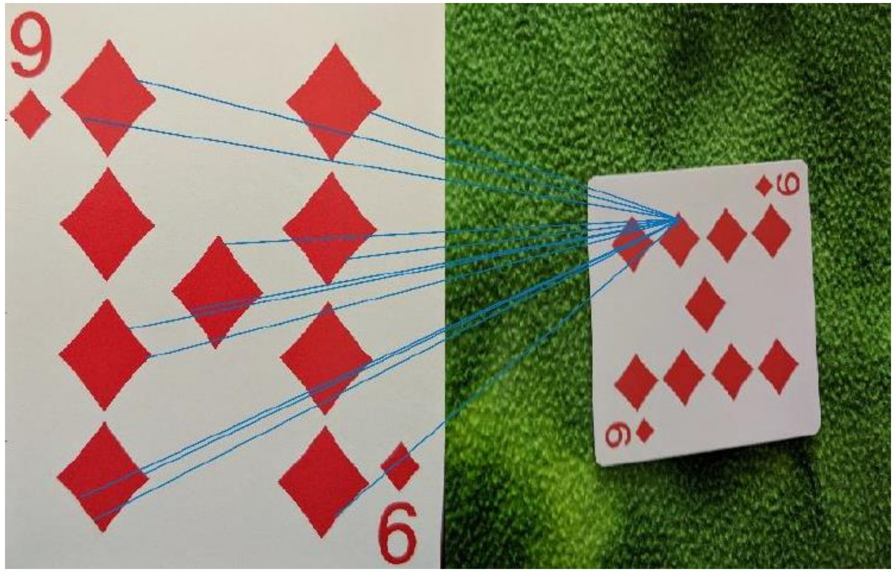


APPROACH 1 : SIFT

- SIFT stands for **Scale invariant feature transform**.
- Why we go for SIFT ? – **Rotation invariance** , **Scale invariance** and **Robust to Lightning**.
- **The main idea** – Matching the SIFT descriptors in the target image with SIFT descriptors that have been precomputed for each type of card.
- The precomputed descriptors are obtained by applying **SIFT keypoint detection** to each of the 52 training images. The location of each keypoint and the corresponding feature descriptor is then stored for later use.
- We can use the **vl_sift** function of MATLAB to find the SIFT keypoints and descriptors. Then for each of the 52 cards , **vl_ubcmatch** is used to match the feature descriptor to the precomputed descriptors. The number of features are then stored.
- If for given card , more than **n** (can be decided) of the features are found in the target image , the card is considered to be in the image.

WHY SIFT FAILED ?

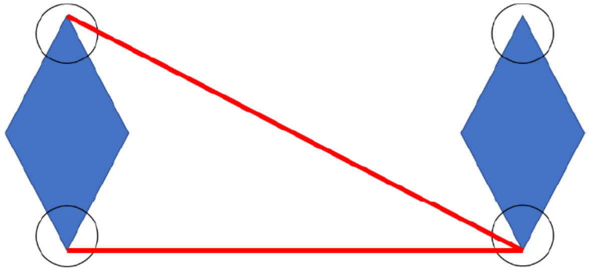
- SIFT provides some inaccuracy in the results . The main reasons why SIFT failed are as follows :
- Multiple features in the training image is matched to the same feature in the target image.



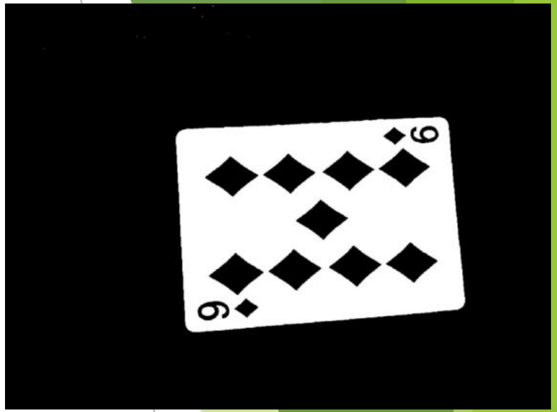
Example of vl_ubcmatch matching multiple features in the template to the same feature in the target image.

WHY DID THE ABOVE HAPPEN ?

- The reason for SIFT facing this issue is because of the highly symmetric nature of playing cards. Playing cards have often have the same symbols repeated multiple times on the card. Due to this symmetry within the card , many features have similar feature descriptors.
- SIFT aligns the feature descriptors by their dominant orientation (like corners of a diamond) , some feature descriptors are almost impossible to distinguish from each other.
- In the figure below , the top and bottom corner of left handed diamond (training image) is matched to bottom corner of right handed diamond (test image) .

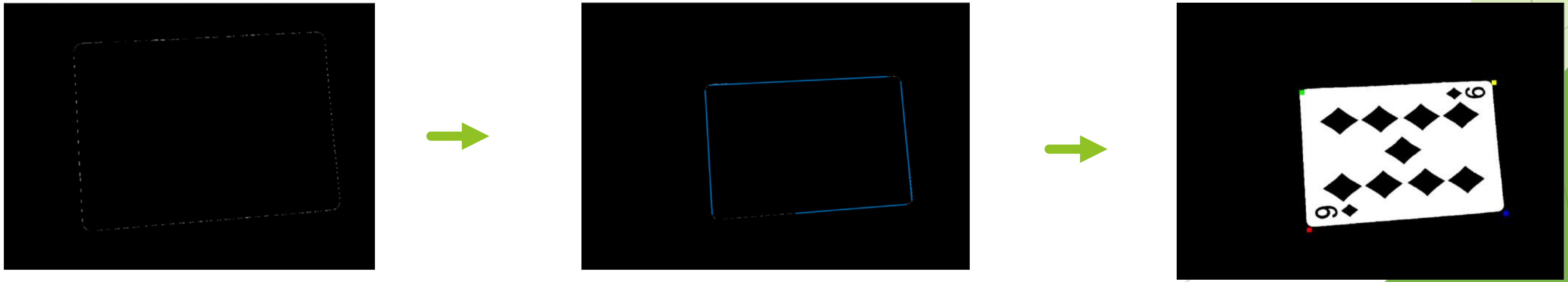


APPROACH 2 : TEMPLATE MATCHING



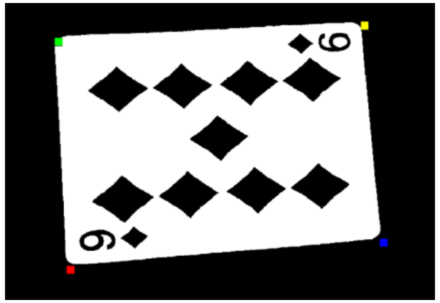
- The steps in this approach are as follows :

- <u>**Threshold**</u> – Card Detection is done via thresholding and the use of MATLAB’s built-in **regionprops** function. In order to apply a threshold , add up the R,G and B values and then a pre-decided threshold is used to binarize the image.
- <u>**Find the Corners**</u> – In order to find the corners : find the edges, apply Hough Transform and then the intersection of lines are corners.



TEMPLATE MATCHING CONTD…

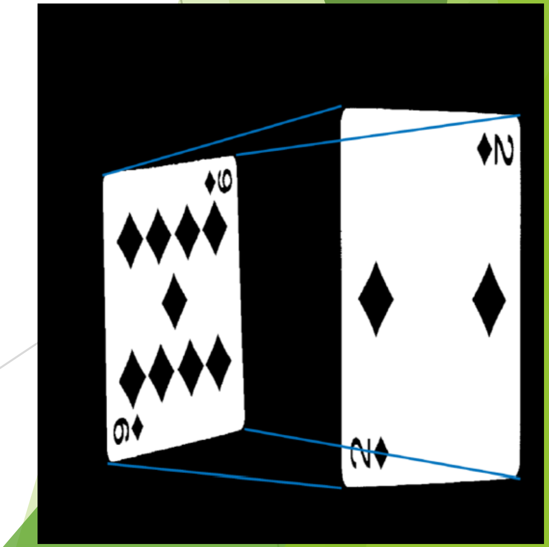
- **Arrange the Corners** – Identify the dark region closest to corner. Mark closest corner as first corner. Go around from there and then rank the corners accordingly.



9 of Diamonds with the corners detected and arranged.

The red corner represents the first corner. The green corner is the second corner. The yellow corner is the third corner, and the blue corner is the fourth corner.

- **Transform Creation and Scoring** – Using four ordered corners in both template and the target image , the homography is found that maps the corners on template onto the corners of the target. Using the transform found, the area of template image containing the card is mapped onto the target image. The SSD (Sum squared difference) is then taken for all pixels within the card to get score of match, and lower score is preferred.

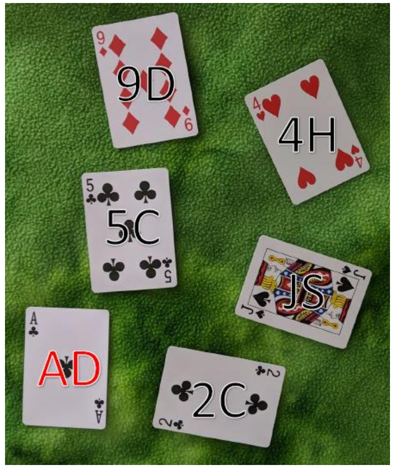


RESULTS OF THE TEMPLATE MATCHING

- Template Matching worked exceedingly well to detect and identify the cards. Despite the high accuracy of the algorithm, there are some flaws.

5/6 of the images

Correctly identified



- Flaws with the Template Matching : It is computationally expensive. The main expense is the scoring part of the algorithm as it runs 52 times for each card, and requires matrix multiplication for every pixel that makes up a card. Subsampling can be utilized to overcome this flaw.

NON-IDEAL CASE PERSPECTIVE DISTORTION

- The algorithms discussed above can work well if the images have a top-down perspective where cards are not distorted.
- For low levels of distortion , the template matching can provide good results but for more extreme levels of distortion , template matching can provide inaccurate results. In figure 1 , there is low level of distortion so template matching can correctly identify the card which is not the case with figure 2.

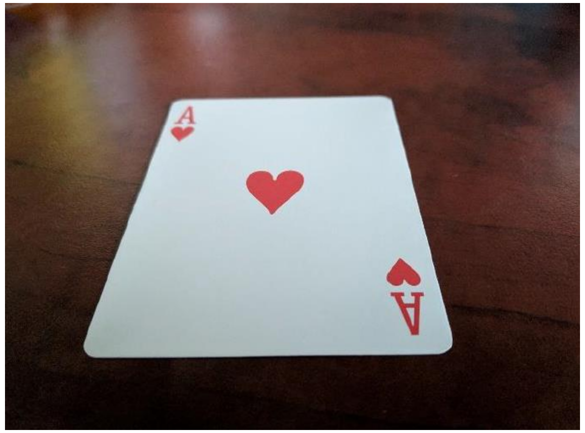


Figure 1

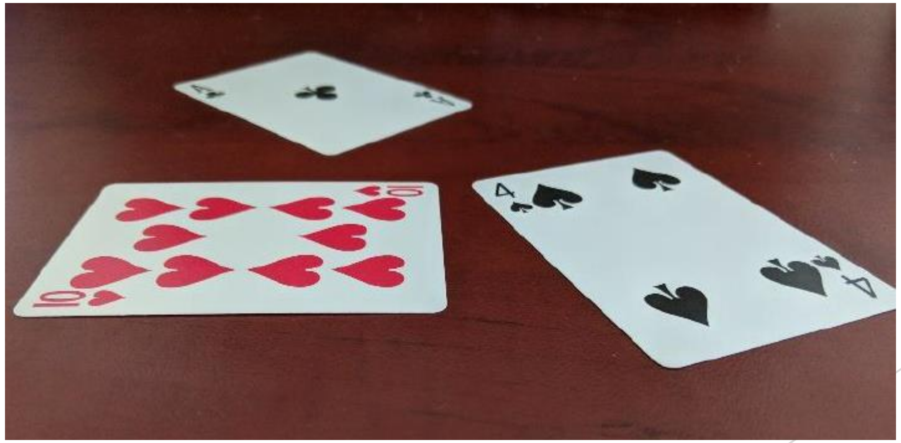


Figure 2

BIBILIOGRAPHY

- (https://web.stanford.edu/class/ee368/Project_Winter_1819/index.html)
- Otsu's method - https://www.youtube.com/watch?v=jUUkMaNuHP8
- https://en.wikipedia.org/wiki/Rotational_invariance
- Dan Snyder – Playing Card Detection : Report
- D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004. Available: 10.1023/b:visi.0000029664.99615.94
- J. Pimentel and A. Bernardino, “A Comparison of Methods for Detection and Recognition of Playing Cards.”

THANK YOU