Group 24
God of Gamblers

Problem Definition

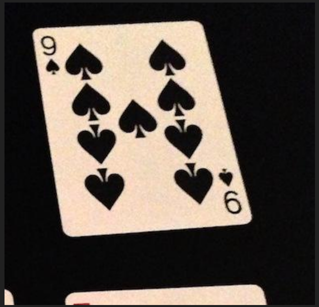
- Identifying playing cards (4 suits × 13 ranks) within a scene
  - Dark background with white cards
  - Cards are not covered
  - These assumptions may be relaxed

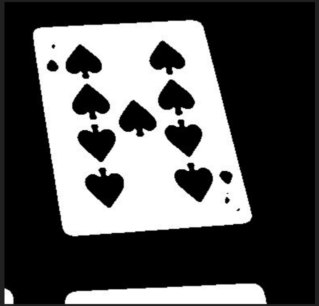
Preprocessing

- Obtain properly oriented card images to build a database
- Extracting the card from the (dark) background
- Locating the perimeters of the card
- Affine transformation
- Filtering the results

Extracting the card from the (dark) background

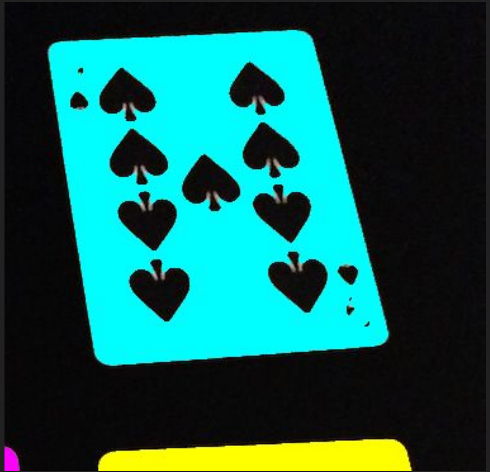
- Grayscale image
- Gaussian smoothing
  - Reduce Gaussian noise while maintaining details
- Thresholding
  - 127
  - Otsu’s method
    - Perform a 2-clustering on all pixels, with centers 0 and 255

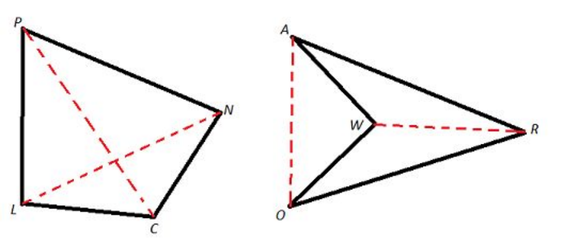




Locating the perimeters

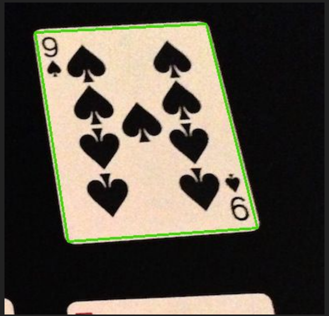
- Connected components
  - In the ideal case, white (object) pixels are separated into components by black (background) pixels
  - A card is simply a connected component
- 2-D Convex hull
  - By definition, a convex hull gives the tightest bounding convex polygon of a set of points
  - A perspectively projected rectangle must be a convex quadrilateral
    - Proof by contradiction: suppose it becomes concave …
- Polygon approximation
  - Reduce the number of corners to 4

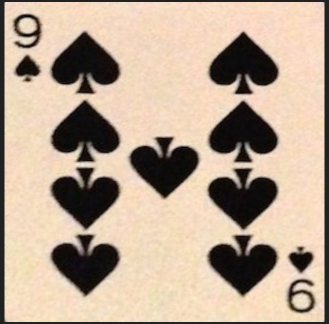




Affine transformation

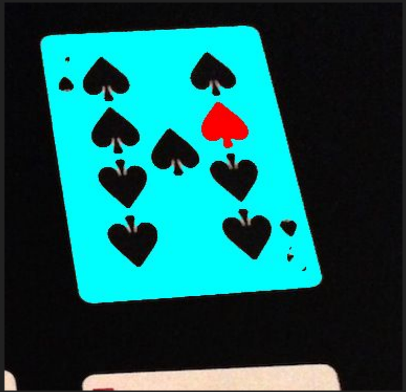
- Converting a perspectively projected rectangular patch to its upright position
    - At this stage we permit a rotated card image - there are at most 4 cases to check, in the recognition stage
    - Now we can create our training database



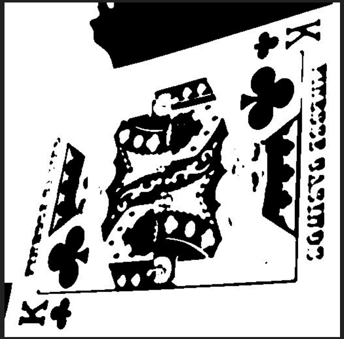


Filtering the results

- Purpose
  - To remove some false positives
- Shape
  - Inertia: how elongated an object is
  - In our program, we limit the aspect ratio of the white patch to between 1 : 0.6 and 0.6 : 1
- Blob detection
  - A card should contain some symbols, which are dark pixels within a white background
  - If a white region does not contain any blobs, it does not qualify as a card
    - Simply check the min/max coordinates
- Components fully inscribed within another



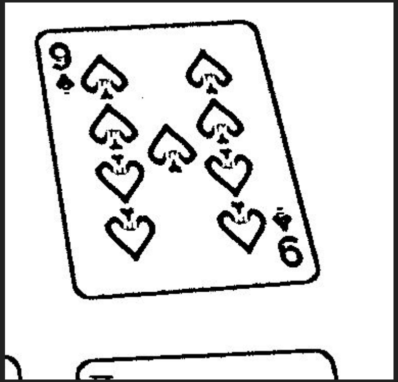
Limitations



- Highly dependent on the thresholding operation
  - If it fails to remove the background, all the remaining computations would fail
  - Runs into similar problems if a card is partially covered
  - Image segmentation is non-trivial
  - Vs. SIFT/SURF: these patented algorithms perform much better as they are agnostic to pixels othe
    interest (ROI)
    - 

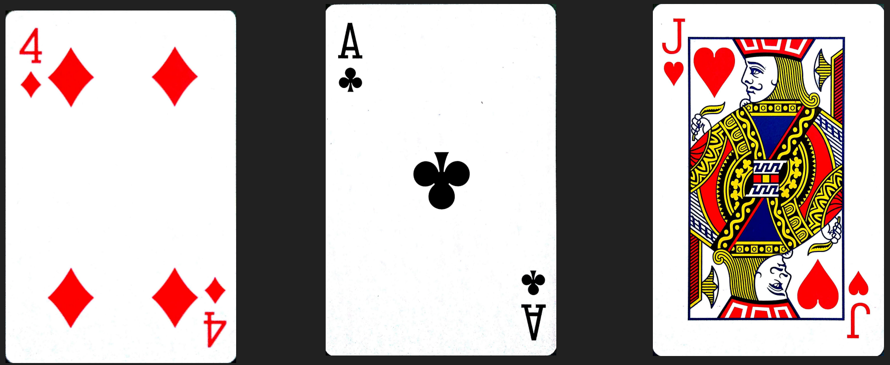
Limitations

- Complexity
  - Current implementation finds the convex hull of all white pixels - $\Omega(n \log n)$ worst-case running time in SciPy implementation
    - n is the number of white pixels in one connected component
    - n can be reduced by operating on the negative Harris corner response image, for instance
      - Or by performing adaptive (local) thresholding
    - Flood fill is $O(n)$



Classifying rank and suit

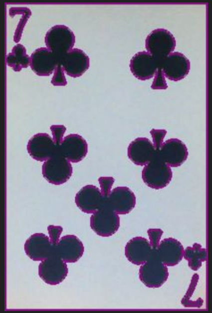
- Problem
- Identifying the rank and suit of a thresholded, upright-positioned rectangular playing card with no background

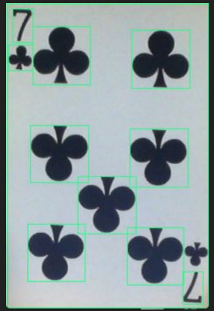


- Solution
- Extract feature from image
    - Position of contour point from x-axis and y-axis
    - 10×10 histogram of highlighted pixel from x-axis and y-axis
    - Wavelet coefficients
- SIFT algorithm

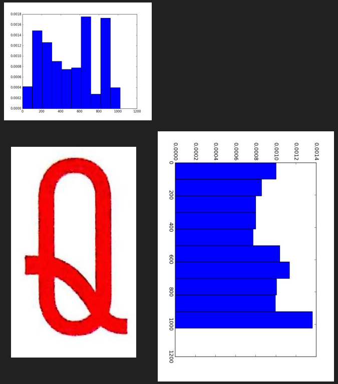
Extract feature from image

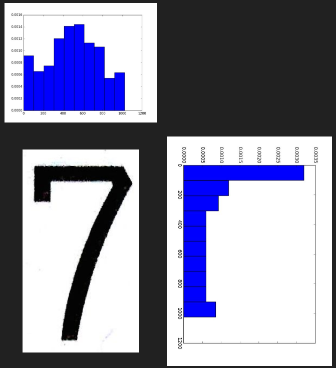
- Identify contours using first principles
- Algorithm:
- Find contours
- Group each contour together with its first layer of children into one
    - Due to the presence of holes in rank “A”, “4”, “6”, “8”, “9”, “Q”
- Crop out for further analysis





Extract feature: Compare contour position

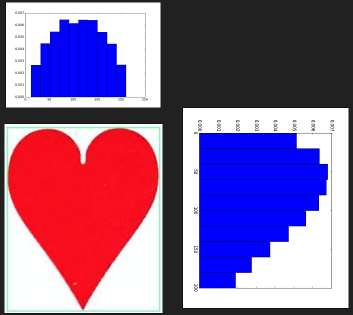


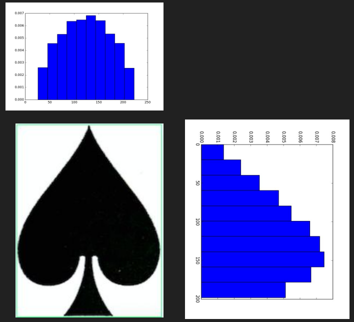


- Compressed cropped area into two list of data
  - Histogram showing the probability distribution of contour points in certain range of x-position
  - Histogram showing the probability distribution of contour points in certain range of y-position
- Compared the histograms of template and test image
  - By correlation
  - By chi-squared
- Limitation
  - Non-rotational invariant
  - Requires high resolution
  - Weak in differentiating object with different “thickness”, e.g. heart and spade in low resolution

Extract feature: Compare highlighted pixel

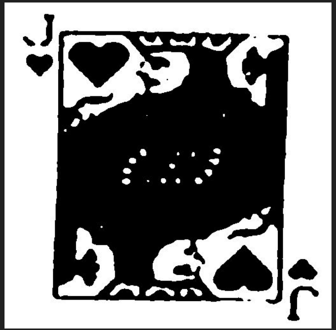
- Compressed cropped area into two list of data
  - Histogram showing the probability distribution of highlighted pixel in certain range of x-position
  - Histogram showing the probability distribution of highlighted ixel in certain range of y-position
- Compared the histograms of template and test image
  - By correlation
  - By chi-squared
- Limitation
  - Non-rotational invariant
  - Requires high resolution





Distortions

- Wavelet compression
  - Works well for digitized signals
  - Separates an image into approximation and detail (c.f. low-pass and high-pass filters in signal processing)
  - Reconstructing an approximate image from less coefficients than the raw image
    - Thereby allowing for some minor distortions
  - What exactly has been compressed?
    - Signals (images) in real life do not tile indefinitely
- Limitation
  - Non-rotational invariant
  - Low separation power among cards with many edges - J, Q, K

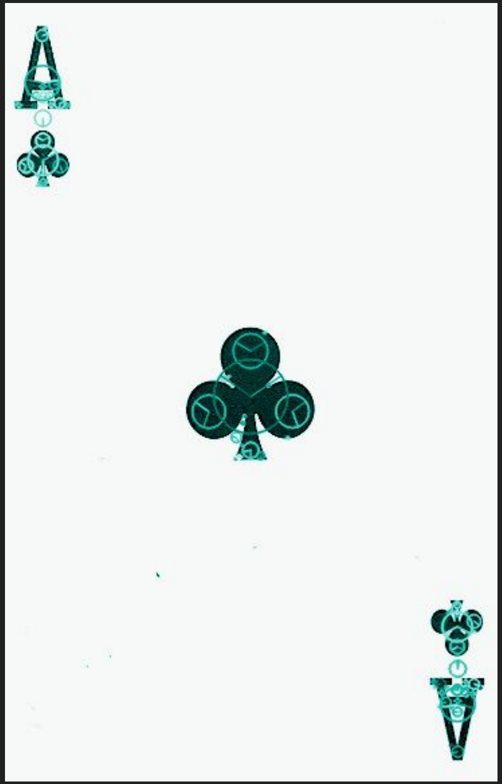


SIFT (Scale-Invariant Feature Transform)

- Another efficient feature extraction algorithm
- First proposed by David Lowe, in 1999
- To find out key-points (with descriptor vectors) of an image
- Robustness
  - Scale
  - Rotation
  - Lighting
  - Noises
- Recognition rate: > 90%

SIFT (Scale-Invariant Feature Transform)

1. Scale space extrema detection
   - Resize image into N different scales (usually N=5)
   - Use Difference of Gaussian (DoG) filter
   - Find local extrema as potential key-points: {x, y, scale}
2. Keypoint filtering
   - Using thresholding, to eliminate low-contrast keypoints
   - Using edge detector, to eliminate edged keypoints
3. Orientation
   - Using Gradient Histogram in 36 directions, to find maxima
   - Get keypoints list: {x, y, scale, orientation}
4. Matching
   - Identifying nearest neighbours in key-points of two images
   - Use Brute-Force matching algorithm



Applications

- Play annotations
- Strategy suggestions