

## **Visión Artificial Clásica – Reconocimiento de Cartas**

Proyecto de reconocimiento automático de cartas de la baraja francesa usando únicamente técnicas clásicas de visión artificial, sin redes neuronales ni algoritmos de aprendizaje automático. El sistema detecta una o varias cartas sobre un tapete verde y reconoce tanto el valor (A, 2–10, J, Q, K) como el palo (♠, ♥, ♦, ♣).

### **Descripción general**

- No se utilizan redes neuronales, SVM ni clasificadores entrenados; todo el reconocimiento se basa en operaciones clásicas de procesamiento de imagen (segmentación por color, contornos, warping, correlación, etc.).
- El sistema funciona con una o varias cartas visibles sobre un tapete verde uniforme, identificando valor y palo de cada carta en tiempo real o desde imágenes estáticas.

### **Características principales**

- Segmentación del tapete verde en espacio de color HSV para aislar la zona relevante de la escena.
- Detección de cartas mediante contornos y aproximación poligonal, asumiendo formas cuadriláteras para los naipes.
- Corrección de perspectiva (warp) para rectificar cada carta a un tamaño fijo (200 × 300 px), facilitando el procesamiento posterior.
- Alineamiento automático de orientación probando rotaciones de 0°, 90°, 180° y 270°.
- Extracción de ROIs (Regiones de Interés) específicas para el valor y el palo en la esquina superior izquierda de la carta rectificada.
- Template matching mediante correlación normalizada para comparar las ROIs con plantillas PNG de valores y palos.
- Modo de calibración interactivo para ajustar las ROIs a cualquier baraja física.
- Reconocimiento estable incluso con cartas de figura (J, Q, K) y variaciones moderadas de iluminación.

### **Hardware**

- Cámara: Webcam o iVCam 1080p.
- Fondo: Tapete verde uniforme (obligatorio para una segmentación sencilla por color).
- Iluminación: Luz ambiental o iluminación adicional homogénea.

El fondo verde permite separar la carta del entorno de forma muy robusta mediante umbralizado en HSV, y la combinación de warping y template matching hace que no se requiera una cámara de alta precisión.

### **Software y librerías**

- Python 3: Lenguaje principal del proyecto.
- OpenCV (opencv-python): Procesado de imagen, segmentación, contornos, warping y template matching.
- NumPy: Operaciones matriciales y manejo de arrays.
- Plantillas PNG: Imágenes de referencia para valores (A, 2–10, J, Q, K) y palos (♠, ♥, ♦, ♣), utilizadas en el proceso de correlación.

### **Flujo del sistema**

1. Captura de imagen
  - Entrada desde webcam o desde un archivo de imagen.
2. Segmentación del tapete verde
  - Conversión de la imagen a espacio de color HSV.
  - Umbralizado mediante `cv2.inRange(hsv, lower_green, upper_green)` para obtener una máscara del tapete.
3. Detección de contornos
  - Búsqueda de contornos en la máscara segmentada.
  - Selección del contorno más relevante con forma de cuadrilátero para cada carta.
4. Warping (corrección de perspectiva)
  - Transformación de la carta detectada a un rectángulo de tamaño fijo de  $200 \times 300$  píxeles.
  - A partir de este punto, todas las operaciones se realizan en coordenadas constantes.
5. Rotación automática
  - Prueba de las rotaciones  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  y  $270^\circ$  sobre la carta rectificada.
  - Selección de la orientación que presenta mayor cantidad de “tinta oscura” en la esquina superior izquierda, asociada a valor y palo.
6. Extracción de ROIs

- ROI de valor (número o figura: 2–10, J, Q, K, A).
- ROI de palo (♠, ♥, ♦, ♣).

#### 7. Template matching

- Comparación de cada ROI con todas las plantillas correspondientes.
- Uso de `cv2.matchTemplate(roi, template, cv2.TM_CCORR_NORMED)` para medir la similitud.
- Selección de la plantilla con respuesta máxima como valor/palo reconocido.

#### 8. Visualización

- Dibujo del contorno de cada carta detectada.
- Overlay de texto con el resultado, por ejemplo:
  - “Q de picas”
  - “7 de diamantes”

### Diagrama de decisión

text

flowchart TD

A[Imagen de cámara] --> B[Segmentación en HSV]

B --> C[Detección de contornos]

C --> D{Hay cartas?}

D -- No --> Z[No se detectan cartas]

D -- Sí --> E[Warp + Rotación automática]

E --> F[Recorte ROIs valor y palo]

F --> G[Template Matching]

G --> H[Seleccionar mejor valor y palo]

H --> I[Mostrar resultados]

### Modo calibración


El modo de calibración permite adaptar las ROIs a cualquier baraja real modificando posición y tamaño de cada región mediante el teclado.

- ROI valor (azul)

- W, A, S, D: mover la ROI.
- Q / E: ajustar el ancho.
- Z / X: ajustar la altura.
- ROI palo (rojo)
  - I, J, K, L: mover la ROI.
  - U / O: ajustar el ancho.
  - N / M: ajustar la altura.
- Controles generales
  - G: mostrar las coordenadas actuales de las ROIs.
  - ENTER: aceptar y guardar la calibración.
  - ESC: cancelar la calibración y volver a los valores anteriores.


## Estructura del proyecto


 Cartas\_IA/

└─  samples/


| └─ values/ → plantillas de números y figuras


| └─ suits/ → plantillas de palos

└─  recognizer.py → núcleo del reconocimiento de valor y palo

└─  detector.py → detección de contornos y warping de cartas

└─  segmenter.py → segmentación HSV del tapete verde

└─  templates.py → gestión de plantillas y template matching

└─  calibrartapete.py → calibración del tapete y ROIs

└─  build\_templates.py → creación automática de plantillas desde cartas reales

└─  config.py → parámetros globales y definición de ROIs

└─  app.py → interfaz principal (webcam/foto)

└─  README.md

## Resultados

- Reconocimiento correcto de todas las categorías de cartas:

- Números (2–10)
  - Figuras (J, Q, K)
  - Ases
- Funcionamiento robusto frente a cambios moderados de iluminación.
  - Reconocimiento correcto en cualquier orientación de la carta gracias al proceso de rotación automática.
  - El modo de calibración permite adaptar el sistema a distintas barajas físicas y tamaños de carta.

## **Conclusión**

El sistema cumple las restricciones académicas al prescindir totalmente de redes neuronales y otros métodos de aprendizaje automático, apoyándose únicamente en técnicas clásicas de visión artificial. La combinación de segmentación por color, warping, rotación automática, ROIs configurables y template matching con plantillas reales proporciona un rendimiento alto y estable en el reconocimiento de cartas de la baraja francesa