

EPFL

EE-451 : Image Analysis and Pattern Recognition

UNO Vision project

**Ecole
Polytechnique
Fédérale
De Lausanne**



PRESENTED BY

Ennouri Hsan, Amine Benzarti and Mokhtar Sellami

What Is the Task?

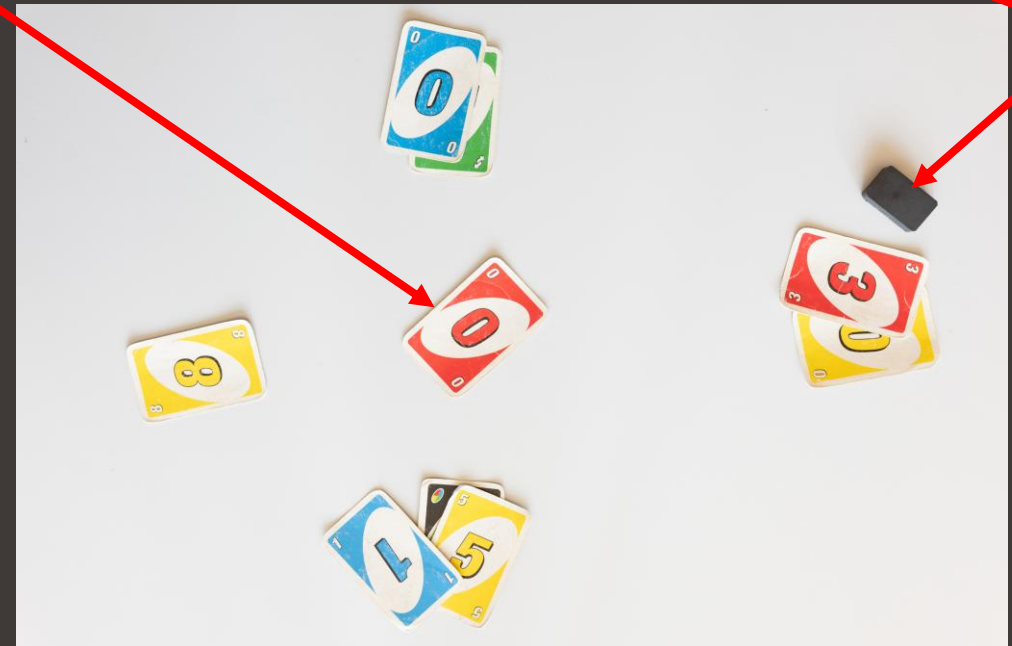
The Task

- Photos of a live UNO game in progress
- Detect for each image:
 - Active player (who holds the token)
 - Cards in each player's hand
 - Card on the center discard pile
- Output: Kaggle-ready CSV submission

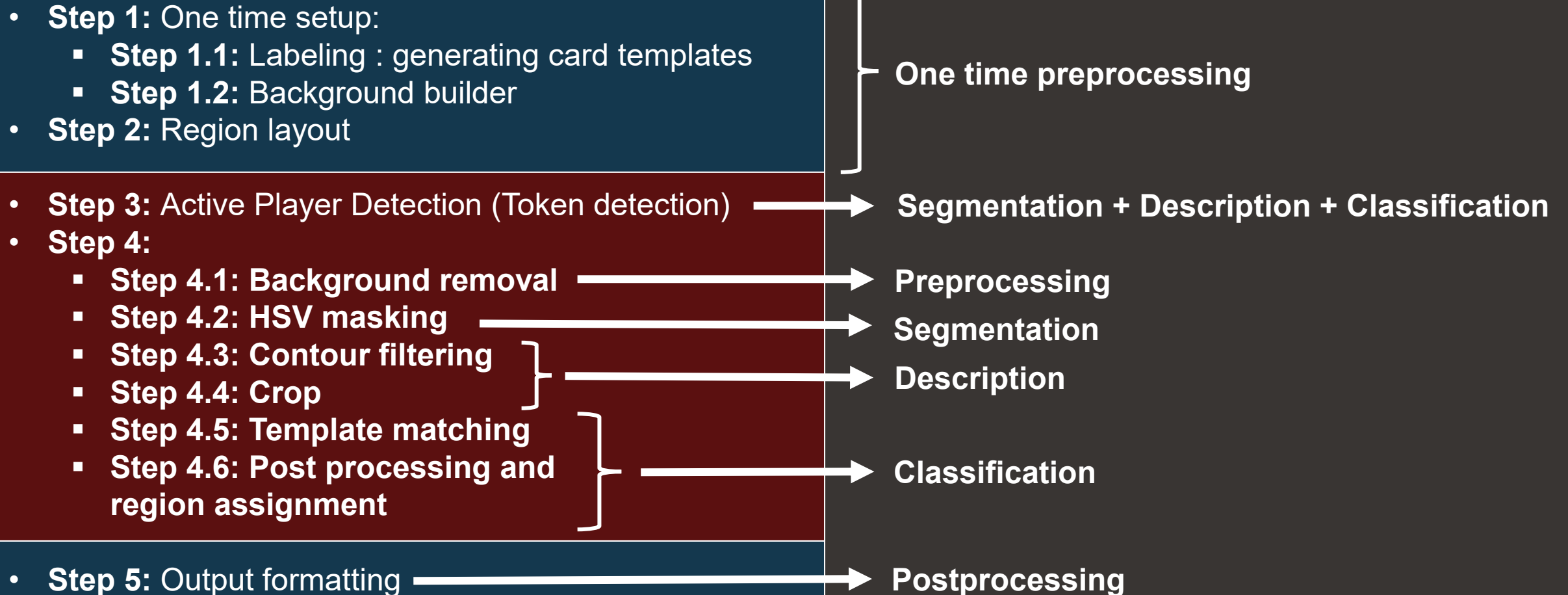
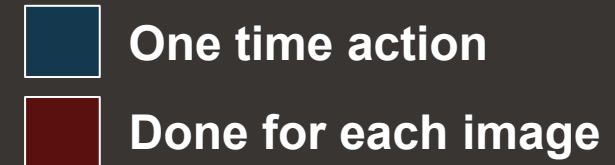
→ We chose a classical image processing approach.

Center
Card

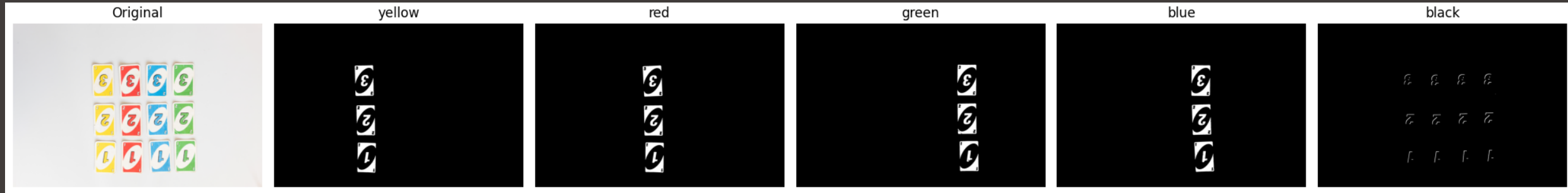
Token



Pipeline Overview



Step 1.1: Labeling : generating card templates



- **Colors:**

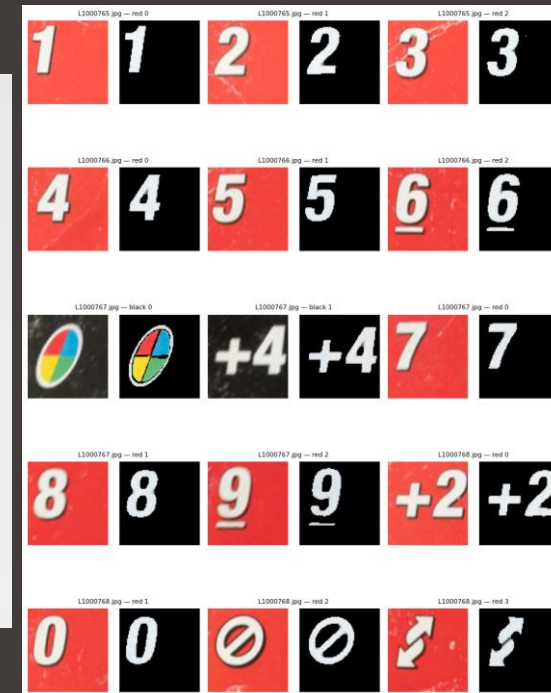
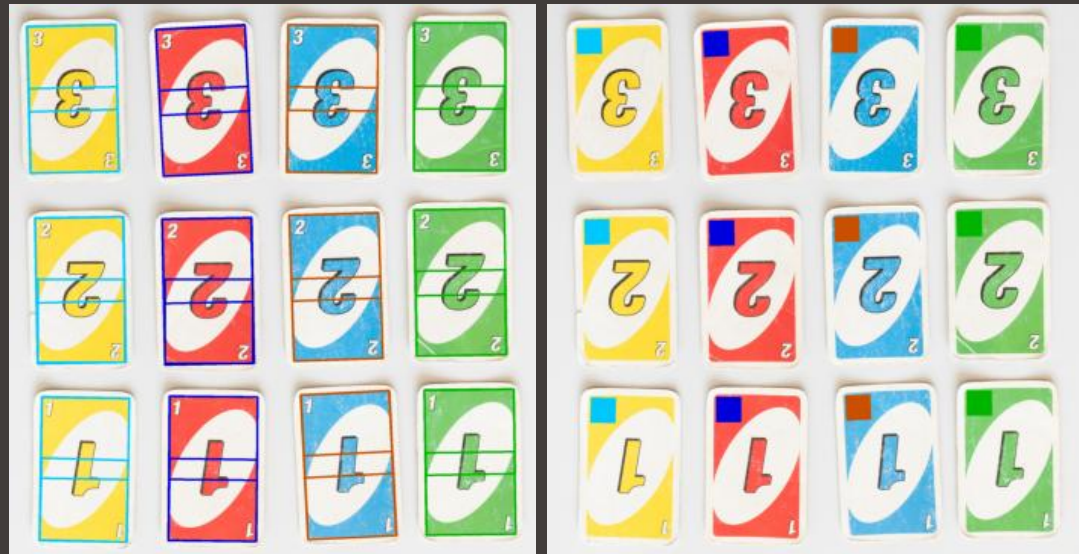
- Fix color ranges for each color (including black)
- Color masking to detect cards

- **Card detection:**

- Generate rectangle boxes containing contours of each color
- Filtering small boxes
- Merging upper + lower part of a card

- **Labeling**

- Get the corner patch of each card
- Mask the color of the card
- Save the templates



Step 1.2: Background builder

- **Background model (only performed once, at the startup)**
- Pixel-wise median over bg training images in BGR
- For each pixel position, keep top 30% stable frames.
- Recompute median and return estimated background



**Blind
Spots**



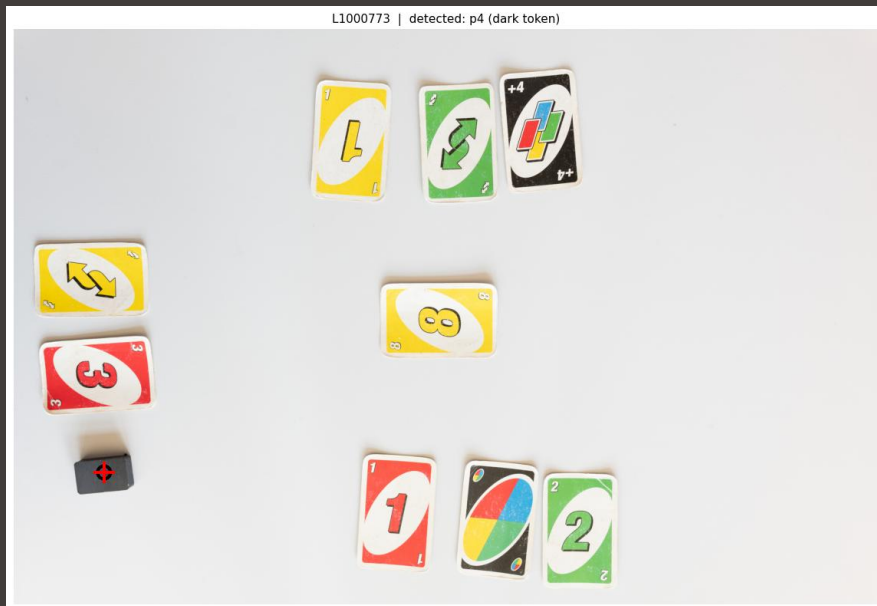
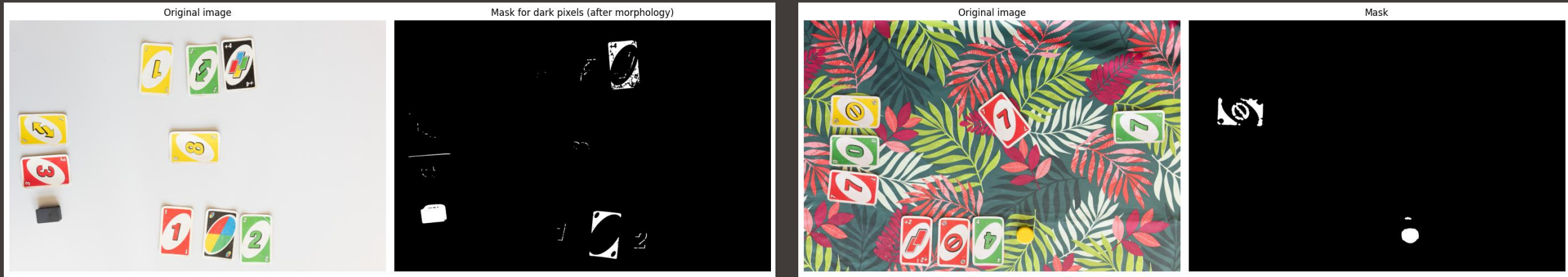
Step 2: Region Layout

How is the Image Split ?

- 5 fixed zones per image:
 - 4 player areas
 - 1 center discard pile
- Boundaries defined as polygons and as fractions of the image size



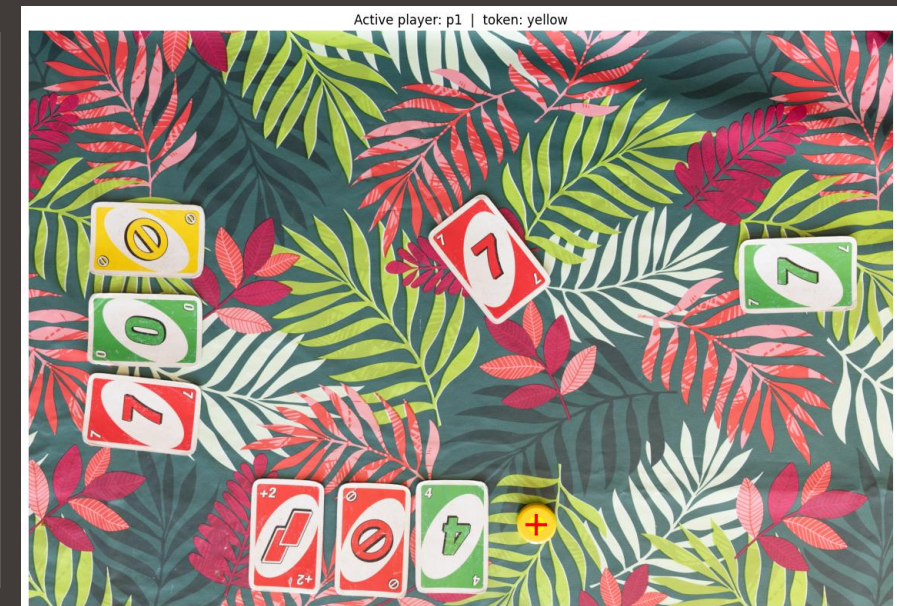
Step 3: Active Player Detection (Token detection)



1. Color mask

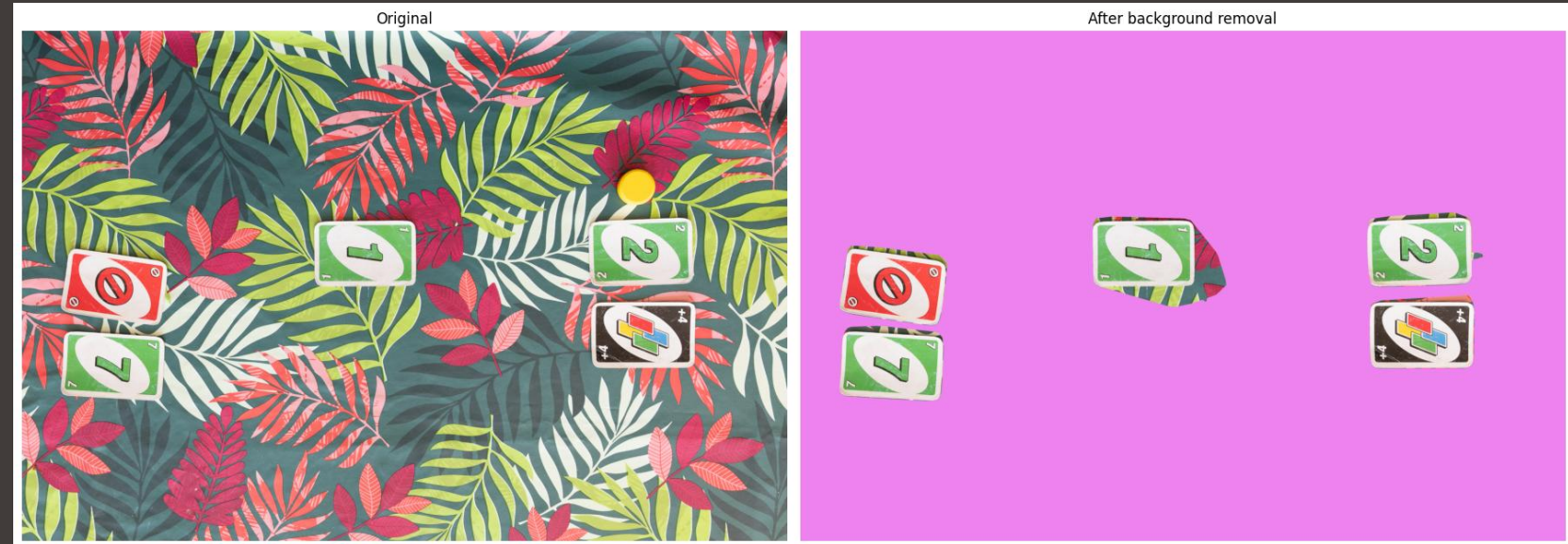
2. Selection Criterias:

- Rectangularity (Black)
- Circularity (Yellow)
- Area
- Aspect Ratio
- Colorful Ratio (Black)

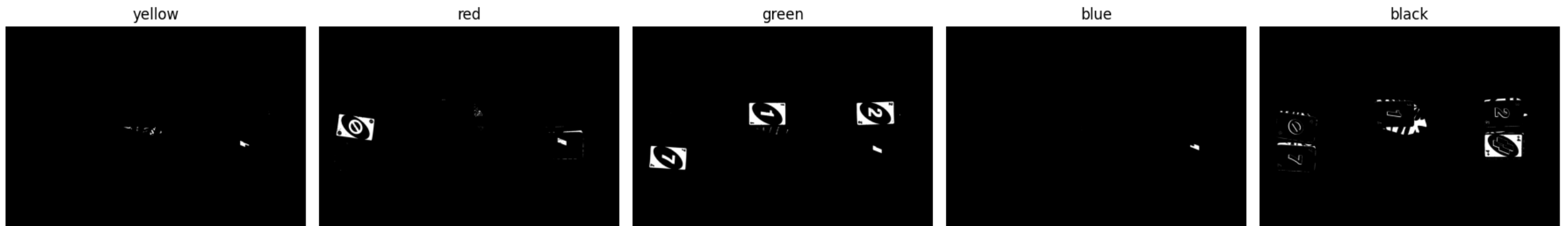


Step 4.1 + Step 4.2 : Background removal (if needed) + HSV Masking

- **Background removal :**
- Compare images in LAB color space and create FG mask
- Refine, filter then apply mask.



HSV color masks after morphological closing



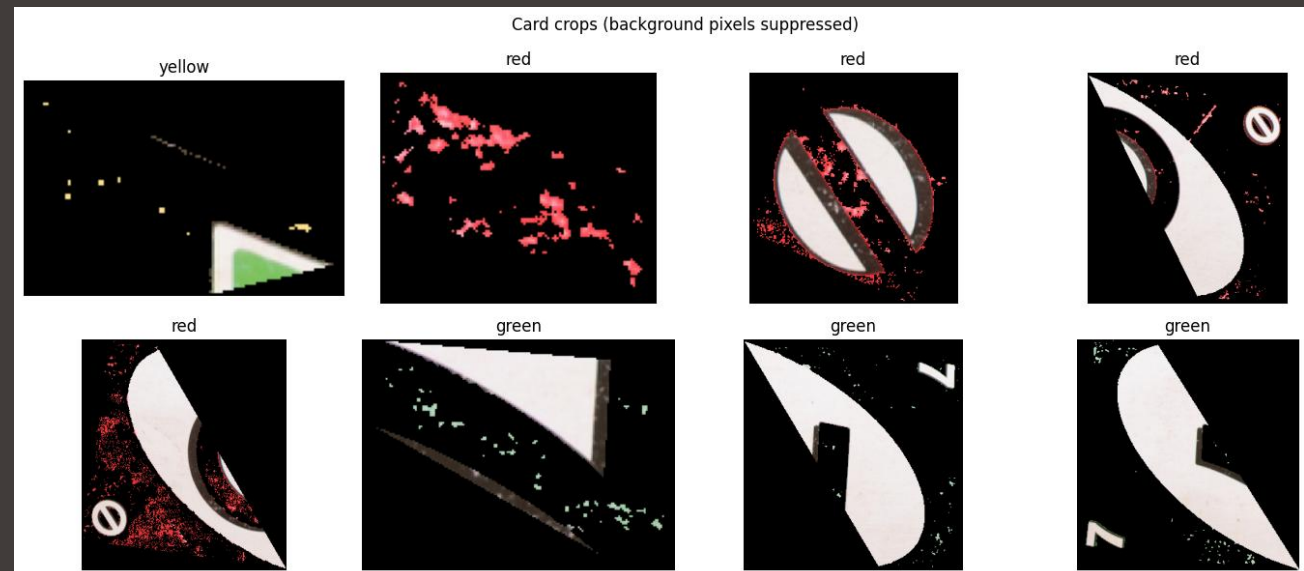
Step 4.3 + 4.4: Contour filtering + Crop

- **Contour Filtering**

- Find the outline of each colored blob
- Keep only card-sized shapes (not too small, not too large)
- green = accepted, red = rejected

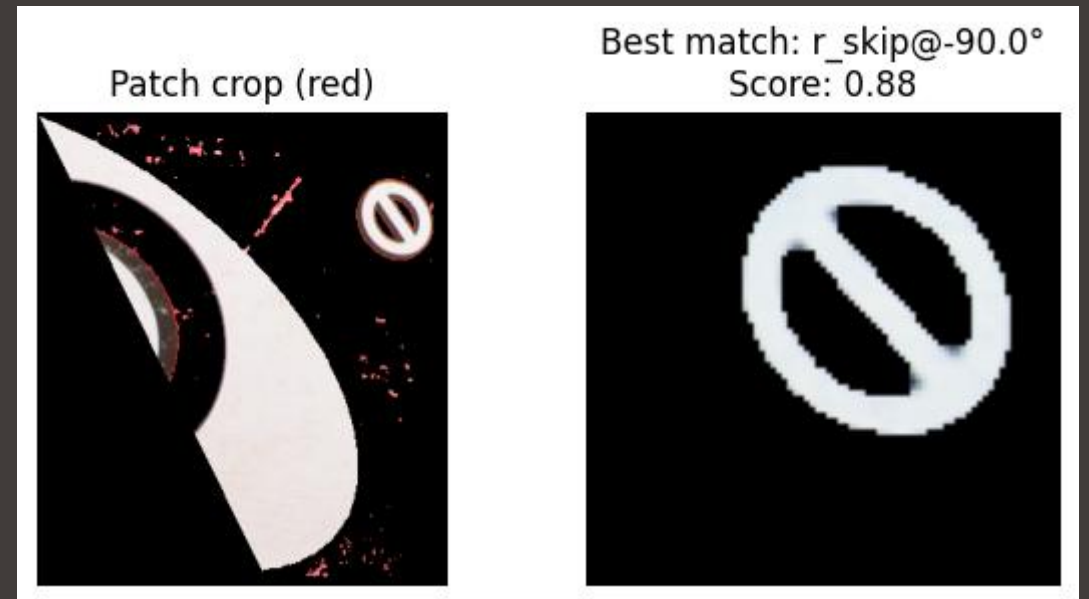
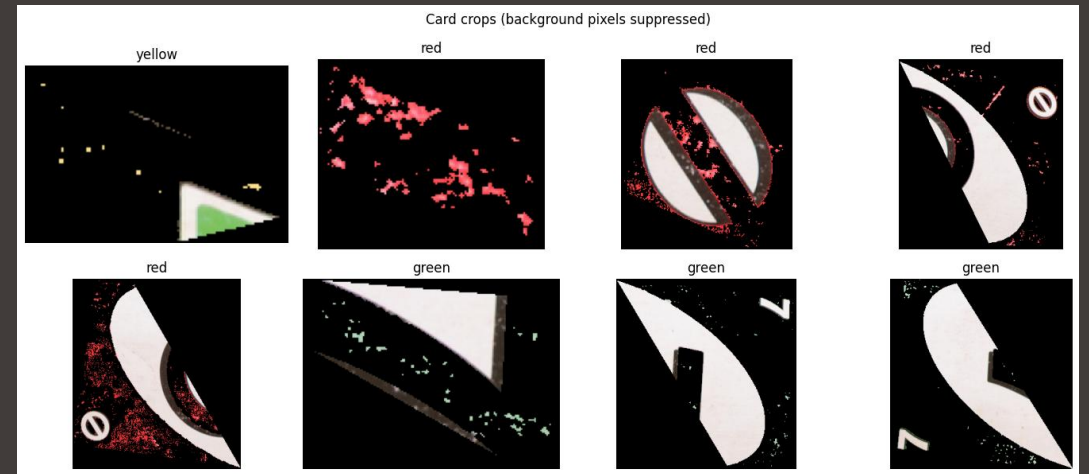
- **Crop**

- Cut out each card from the image
- White masking (not for black cards) → keep only the card
- One patch per card



Step 4.5 Template matching

- **Template matching**
- TM_CCOEFF_NORMED : Lightness and contrast independence
- 20 rotation angles $\rightarrow$ rotation-invariant
- Score threshold ≥ 0.75 (to be considered a candidate)



Step 4.6: post processing and region assignment

Card identification

• Post-processing

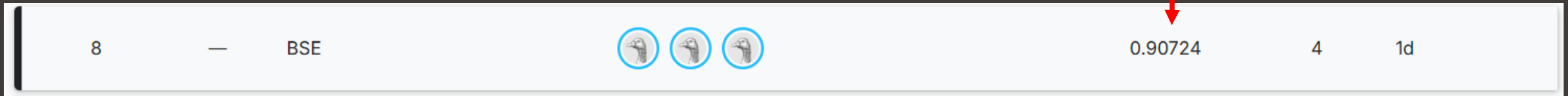
- Threshold tracking: two highest score: draw_2 vs plain 2 → draw_2 preferred
- reverse → higher threshold needed
- Black cards → smaller threshold needed
- Center region: only keep one card(highest score)

• Region assignment

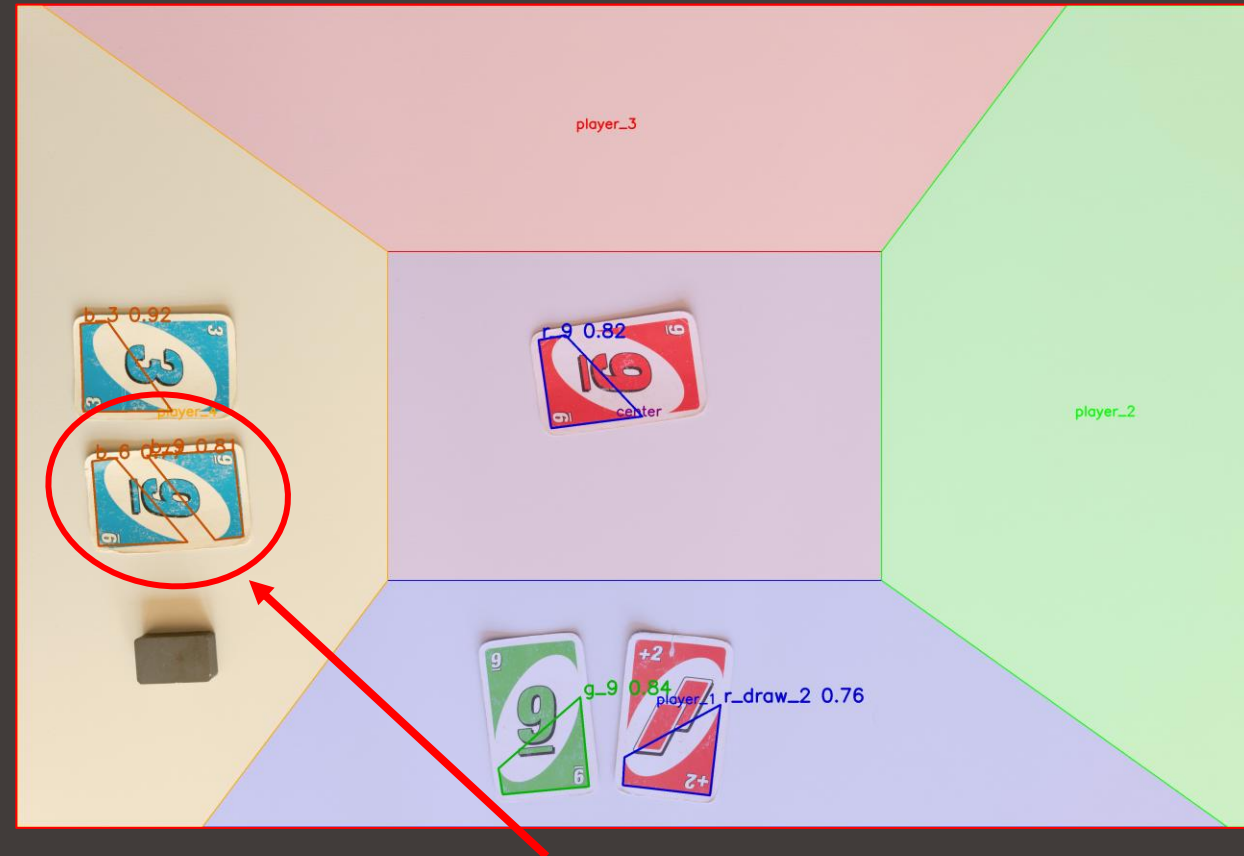
- One card type per region
- Card location defined by the center of the polygone



Limitations (1)

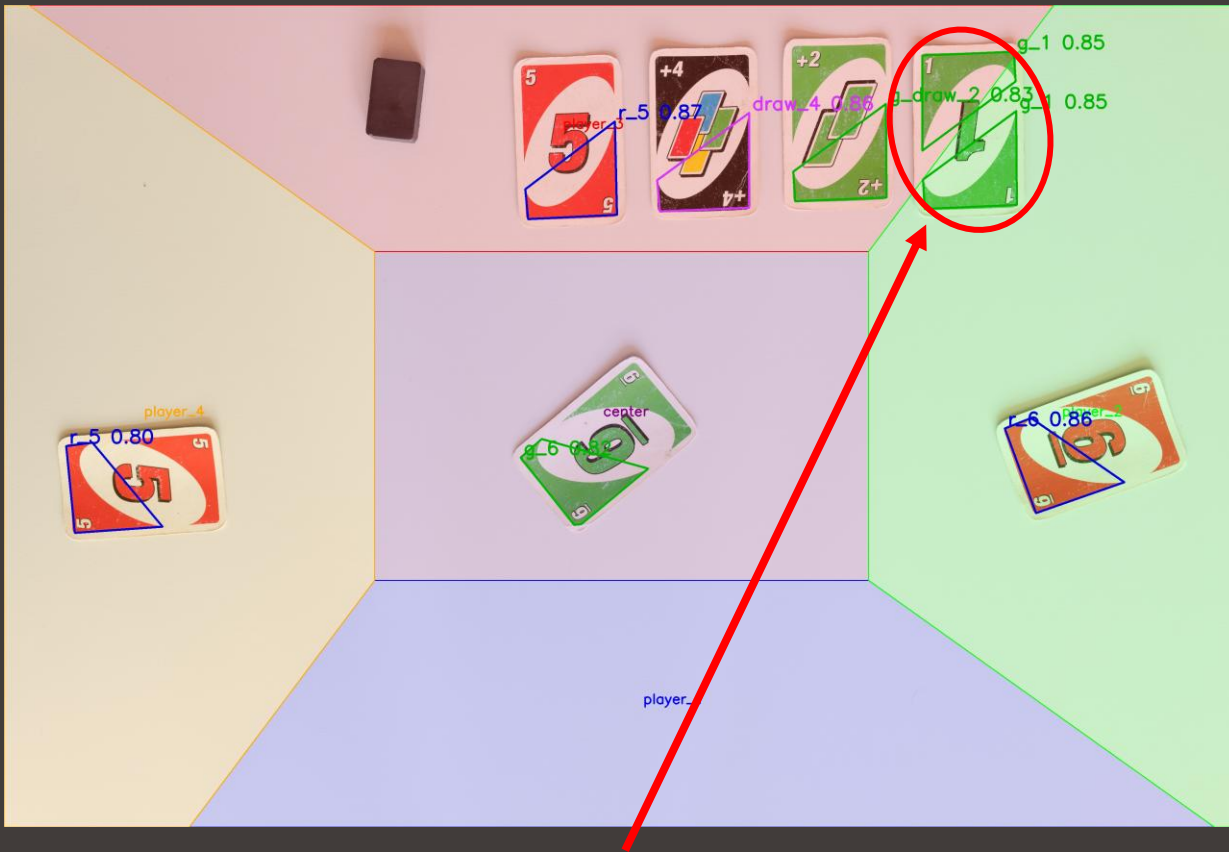


7 VS 1 differentiation



Double detection

Limitations (2)



Region overflow



Some misses