

ELABORAZIONE DELLE IMMAGINI UNO!

Riconosci e beccati sto +4

INDICE

01 INTRODUZIONE

02 ANALISI DATASET e ASSUNZIONI

03 OVERVIEW GENERALE

04 SEGMENTATION

05 OBJECT DETECTION

06 OBJECT RECOGNITION

07 CARD CLASSIFICATION

08 ANALISI DEI RISULTATI e CONCLUSIONI

09 CONTRIBUTI

10 APPENDICE

UNO

O1

INTRODUZIONE

INTRODUZIONE

Obiettivi del Progetto:

- Riconoscimento e classificazione carte “Uno!”
- Identificazione di gruppi di carte
- Identificazione di oggetti estranei





02

ANALISI DATASET e ASSUNZIONI

ANALISI DATASET

- Descrizione delle principali **caratteristiche** delle immagini
- Scelta di immagini “**training**”



presenza di **riflesso di luce**



presenza di **oggetto estraneo**



presenza di **scarsa luminosità**



aggiunta di **cerchio rosso**



aggiunta di **penna gialla**



aggiunta di **macchia gialla**

ANALISI DATASET

Quali sono le immagini che potrebbero portare più **problem**i?



img 16

presenza del quaderno giallo



img 17

presenza dei gruppi di carte

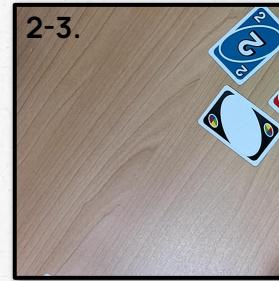


img 24

forte luminosità sulle carte "back"

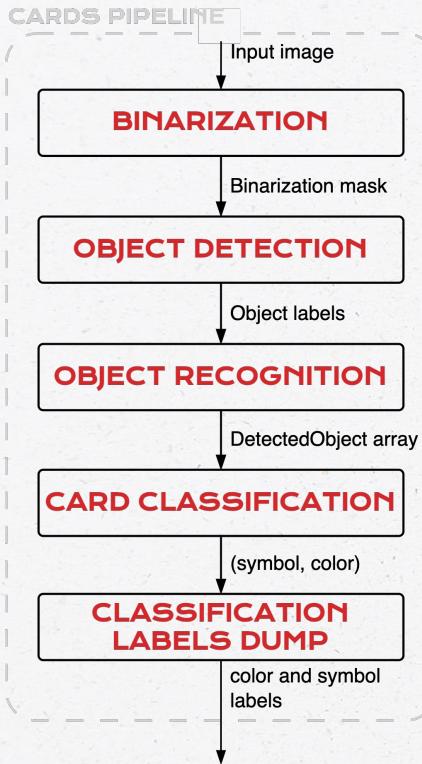
ASSUNZIONI

- 1) Dato un gruppo di **carte sovrapposte**, definiamo come "estraibili" solo quelle dove si vede l'ovale e come "riconoscibili" le carte con anche i simboli agli angoli visibili.
- 2) Supponiamo che tutte le immagini abbiano il medesimo **sfondo** marroncino con venature di legno.
- 3) Assumiamo che la **luminosità** delle immagini sia sempre simile a quella data dal DataSet.
- 4) Definiamo una carta come "**back**" se ha forte presenza di nero negli angoli e colori rosso-giallo in centro alla carta.
- 5) Affinché una carta venga identificata come "**back**", assumiamo che non debba essere soggetta a riflessi di luce molto forti ed omogenei.
- 6) Un **oggetto estraneo** viene definito come tale se ha dimensione inferiore alla carta secondo un determinato fattore (riguardante il l'esclusione dei riflessi).
- 7) Supponiamo che gli **oggetti estranei** presenti nelle immagini siano fortemente diversi dalle carte per forma o colore.

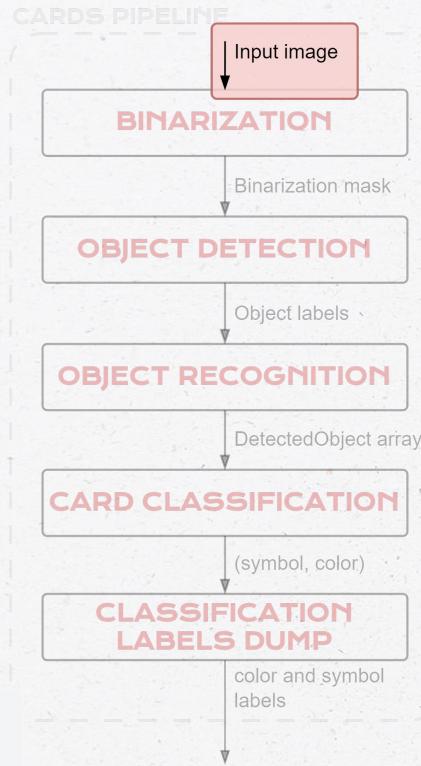


03

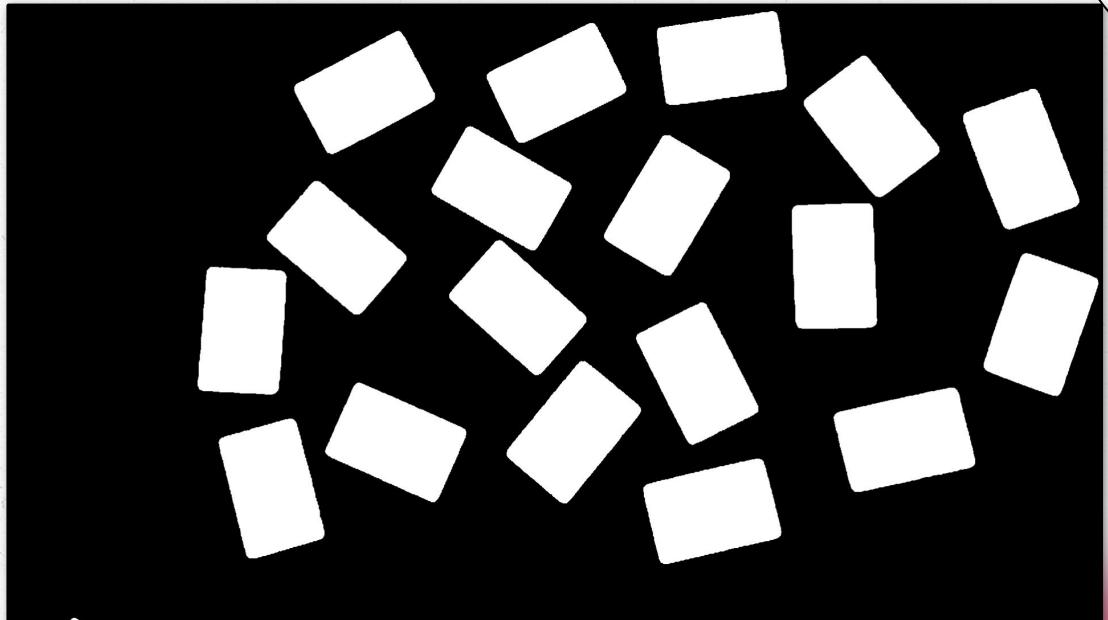
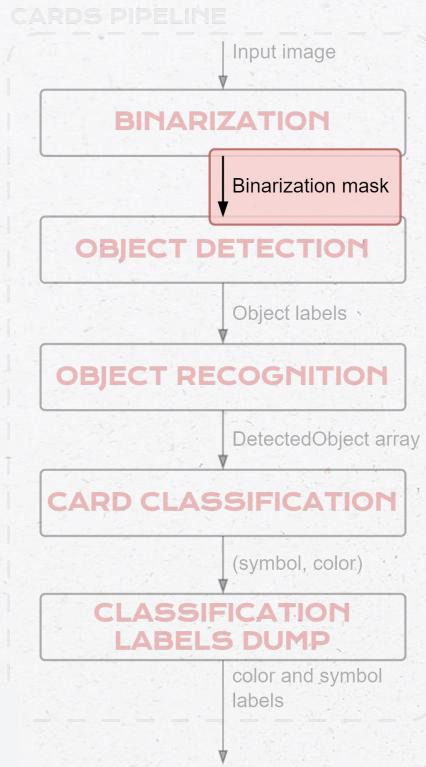
OVERVIEW GENERALE



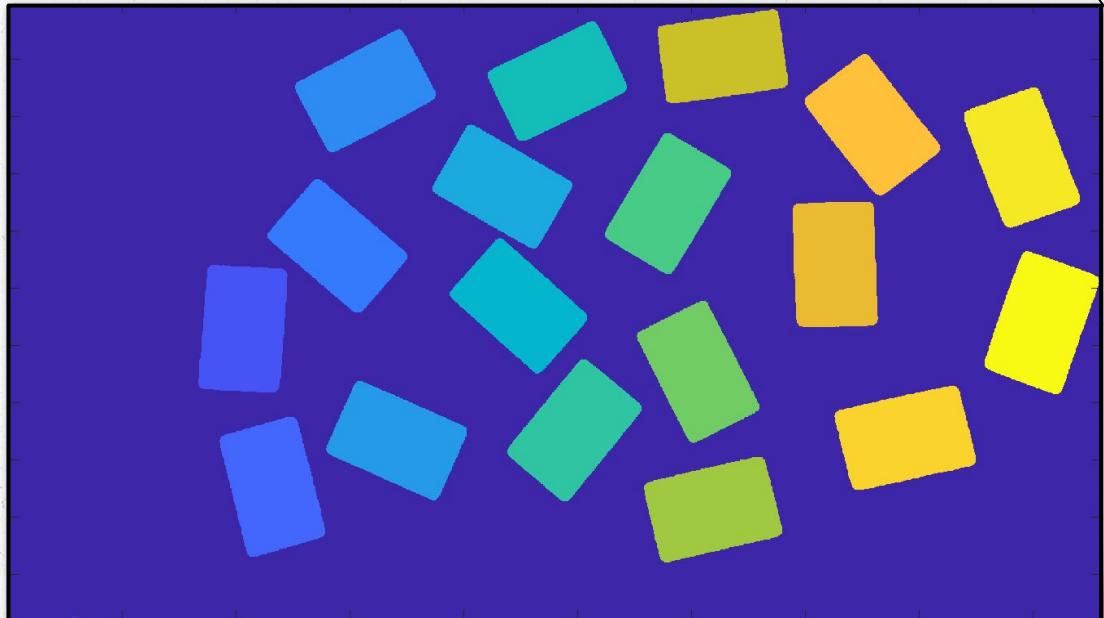
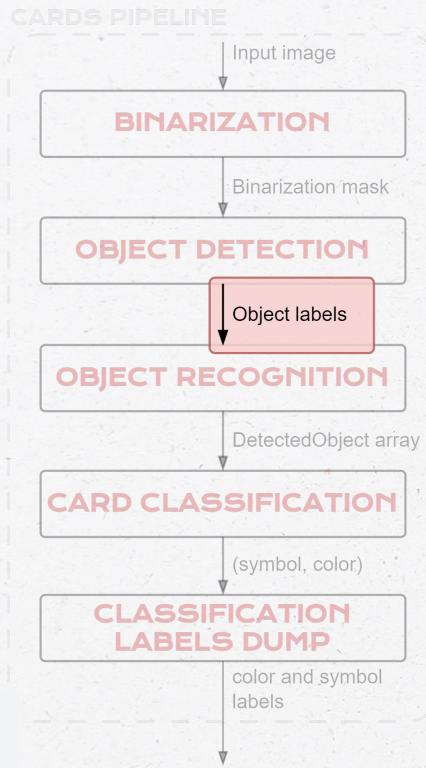
PIPELINE MACRO MODULI



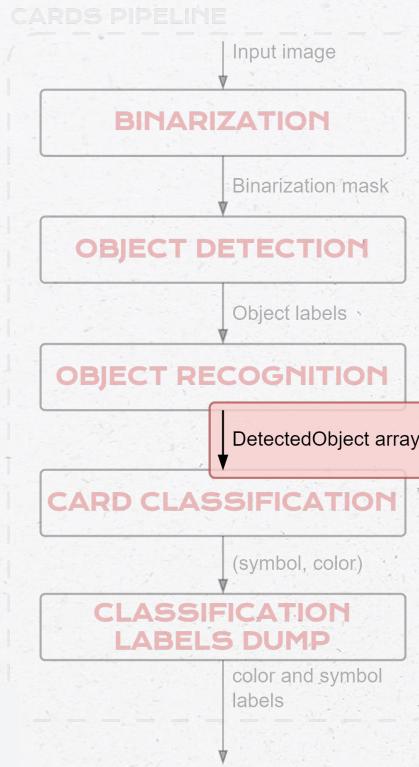
PIPELINE MACRO MODULI



PIPELINE MACRO MODULI

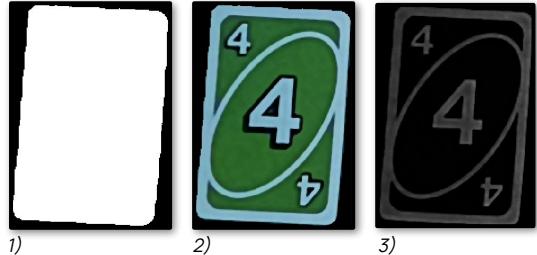


PIPELINE MACRO MODULI



DetectedObject:

1. BW
2. RGB
3. V
4. MASK
5. Symbol: *undefined*
6. Color: *empty*

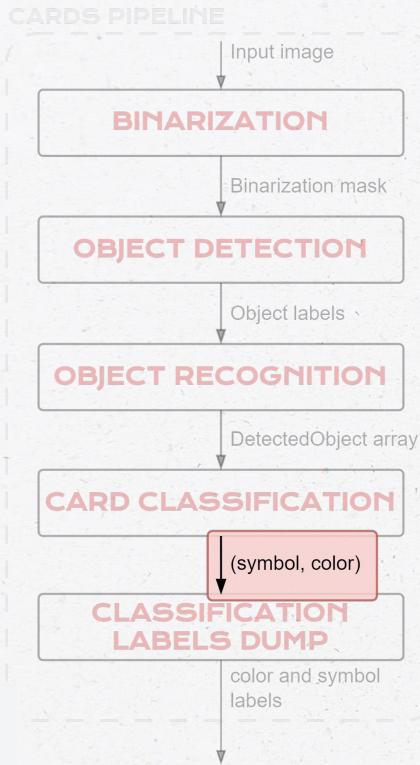


1) 2) 3)



4)

PIPELINE MACRO MODULI

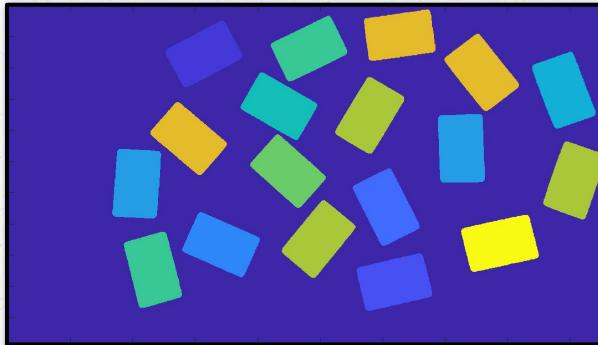


DetectedObject:

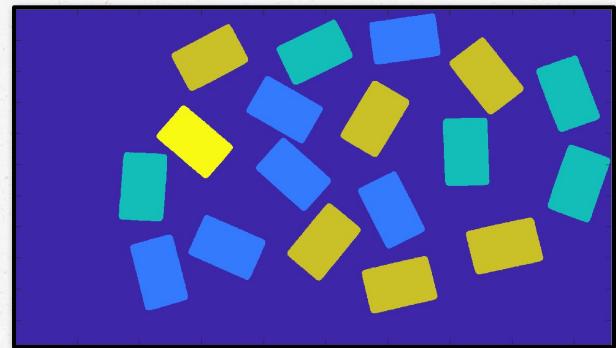
1. BW
2. RGB
3. V
4. MASK
5. Symbol: "*seven*"
6. Color: "*yellow*"

La classificazione imposta gli attributi Symbol e Color di ogni DetectedObject pari ai valori predetti

PIPELINE MACRO MODULI



Labels classificazione **simboli** carte



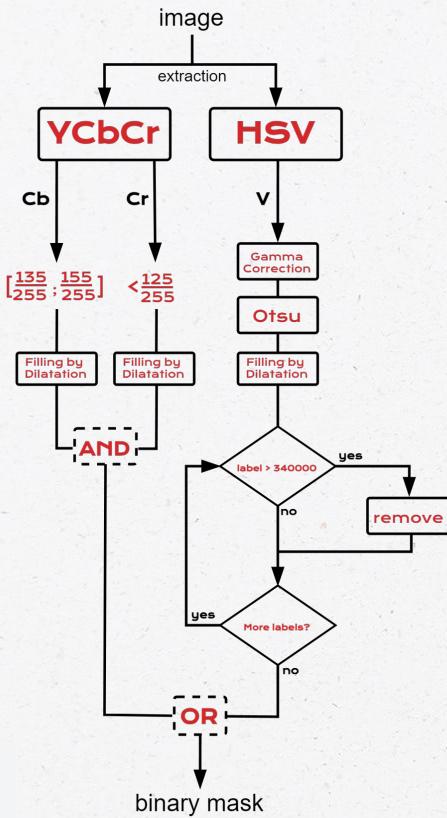
Labels classificazione **colore** carte



04

SEGMENTATION

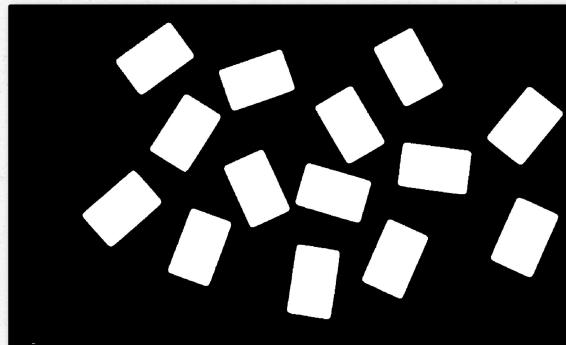
SEGMENTATION



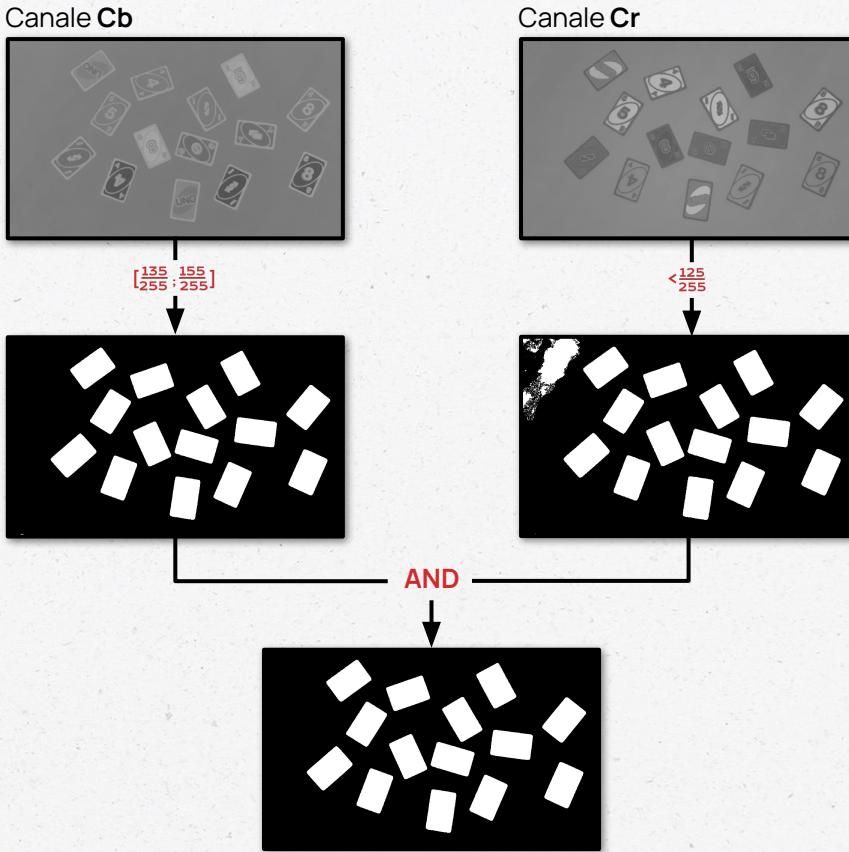
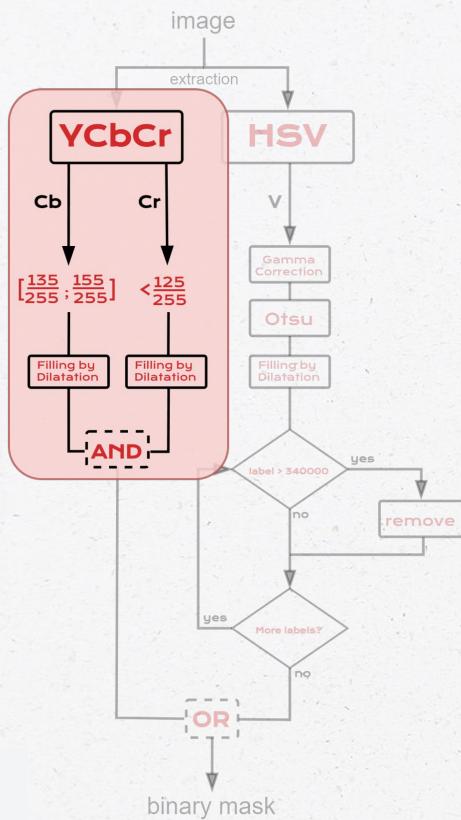
From:



To:

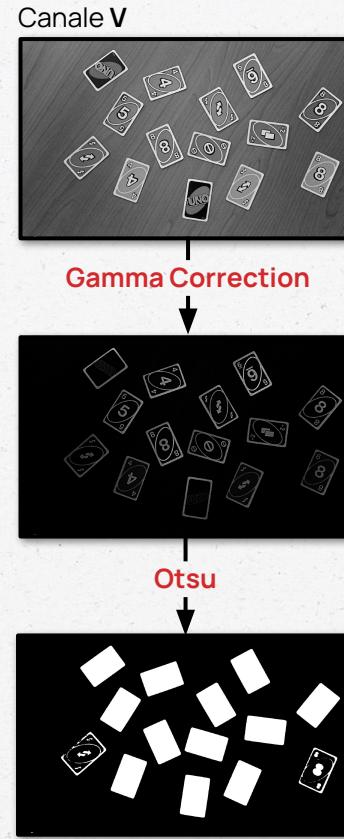
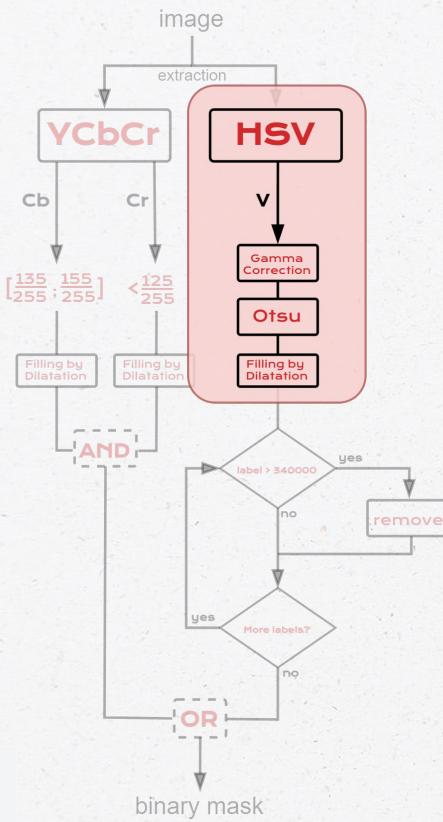


SEGMENTATION

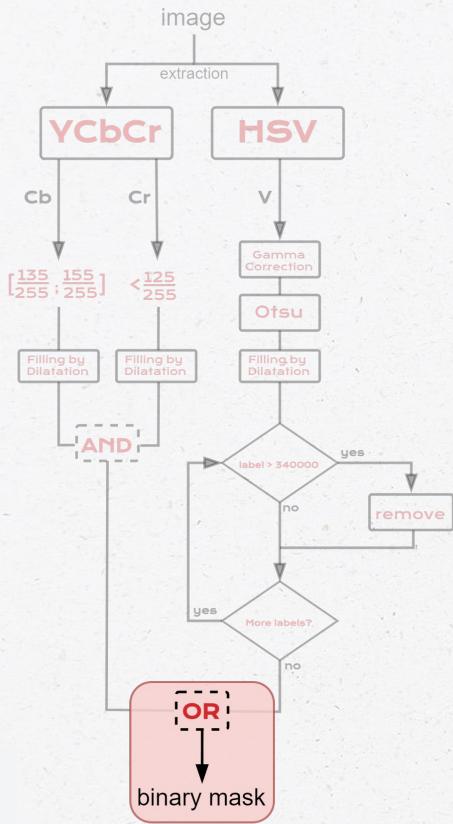


Combinazione Cb-Cr binarizzate

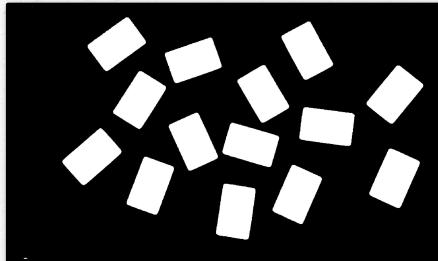
SEGMENTATION



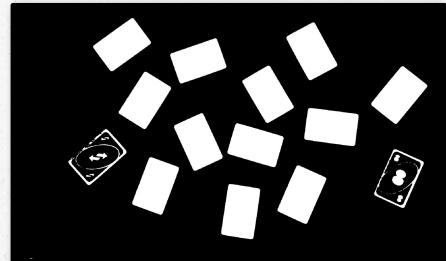
SEGMENTATION



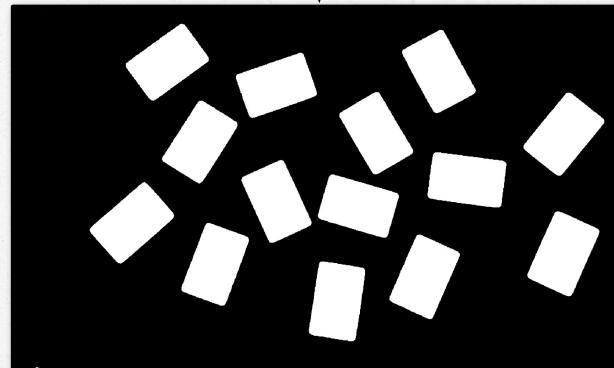
Cb-Cr mask



V mask

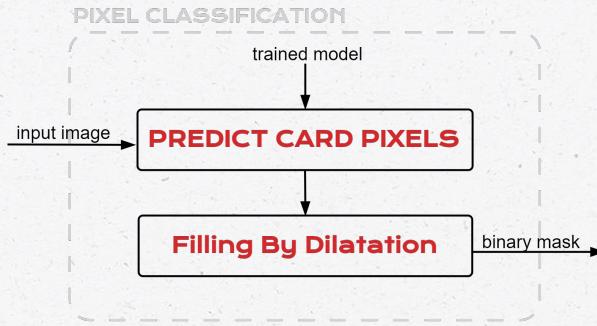


OR



Final Binary Mask

Alternative Considerate bg pixel classification

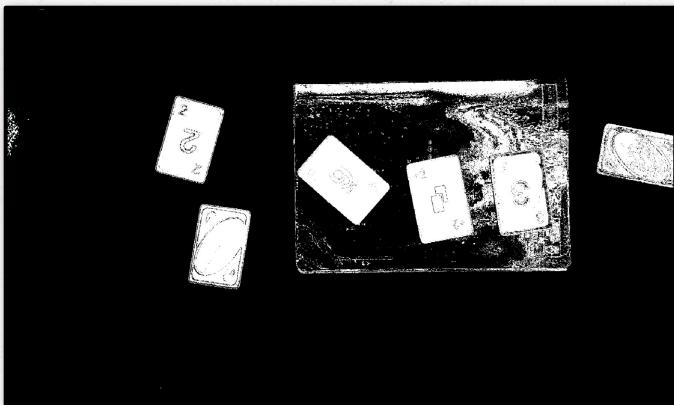


Problem:

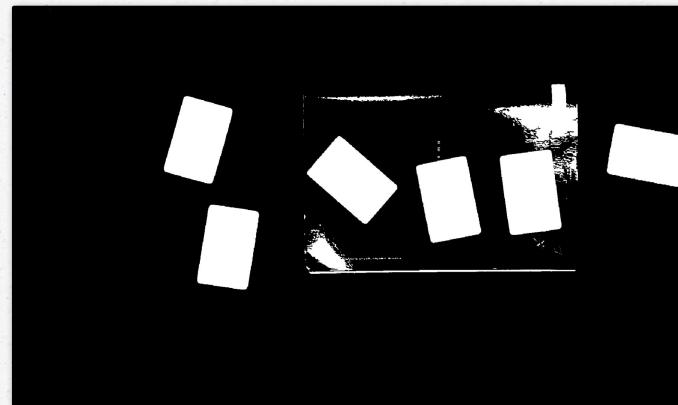


Alternative Considerate bg pixel classification

Confronto:

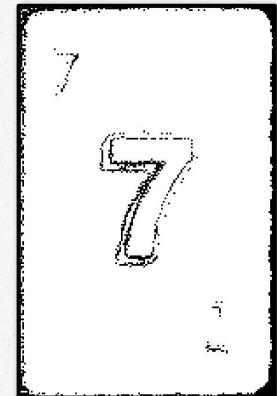
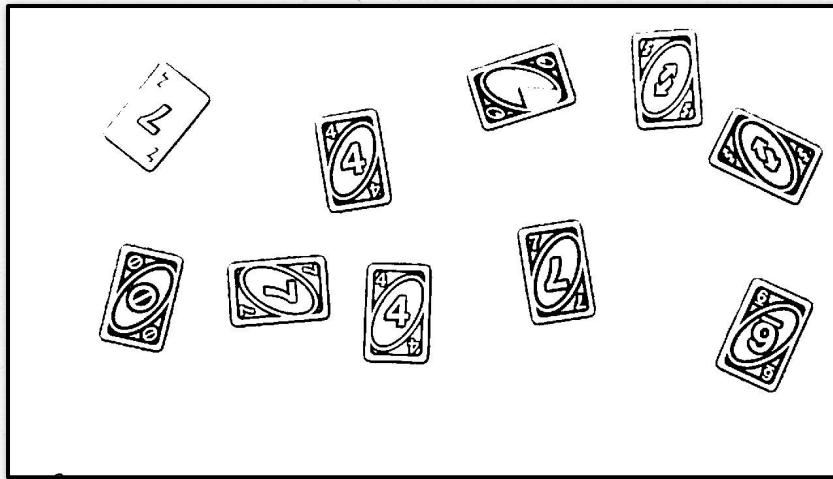
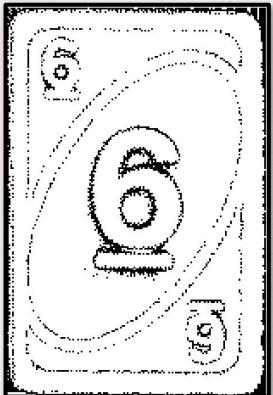


binarizzazione con **classificazione pixel**



binarizzazione con **modellazione canali colore**

Alternative Considerate sauvola



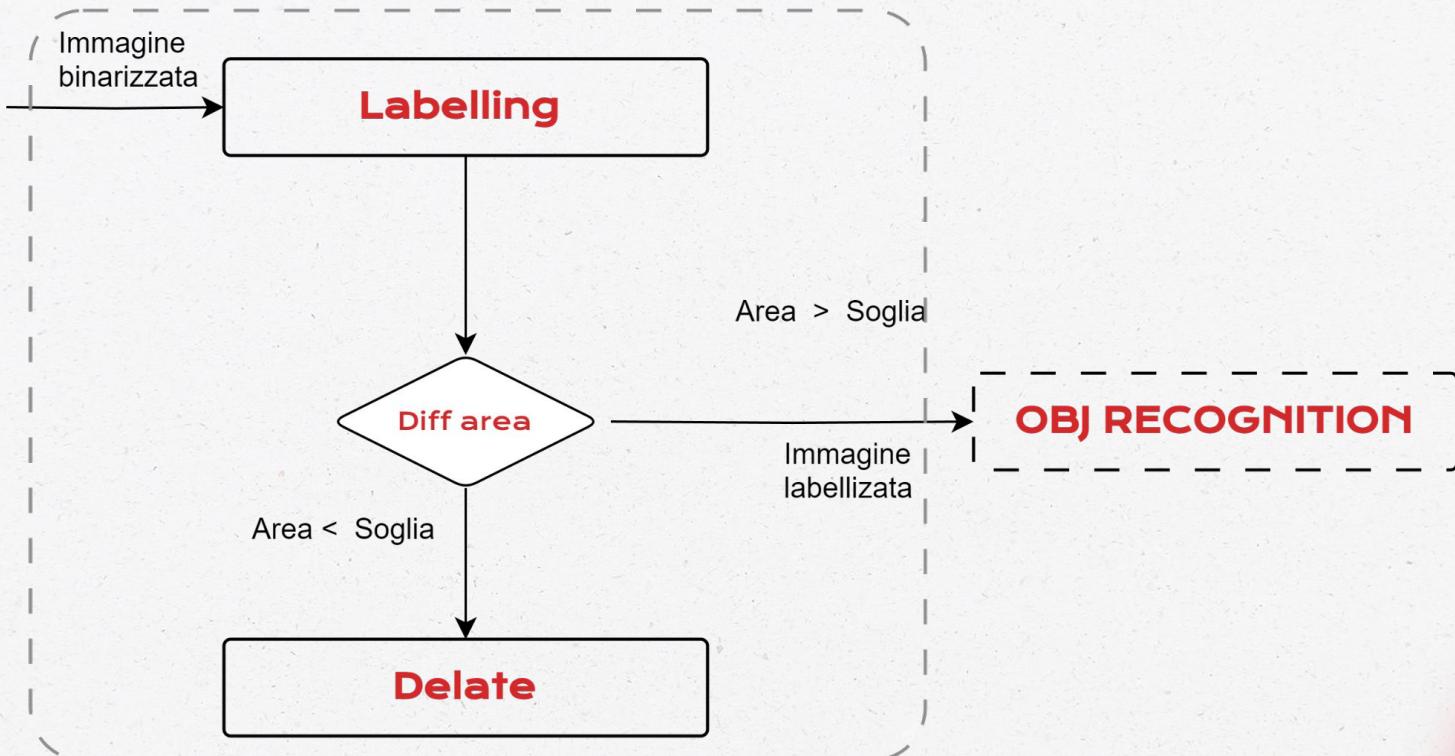
Problema soprattutto con le carte di colore giallo



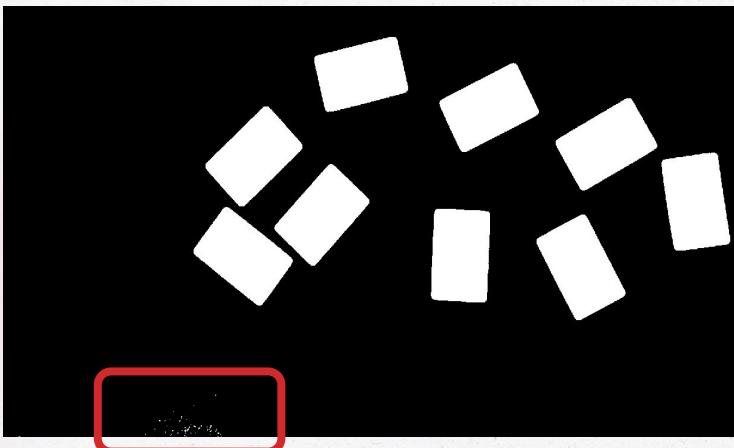
05

OBJECT DETECTION

OBJECT DETECTION

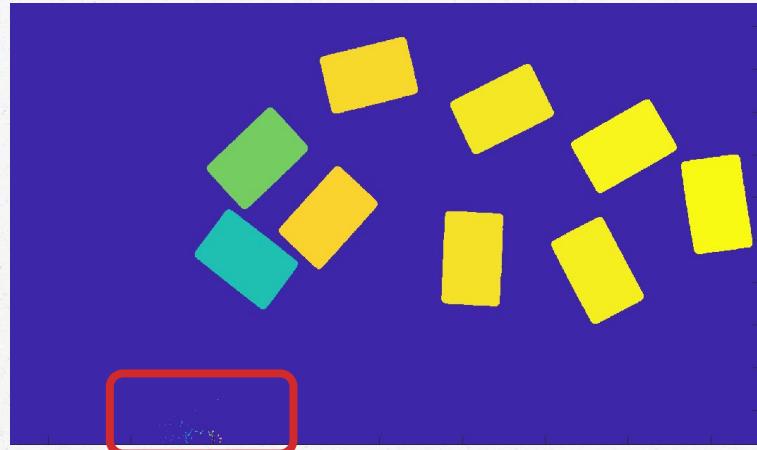


FUNZIONAMENTO



Area media di una carta = $3.0081e+04$ px

Soglia = Area / 5



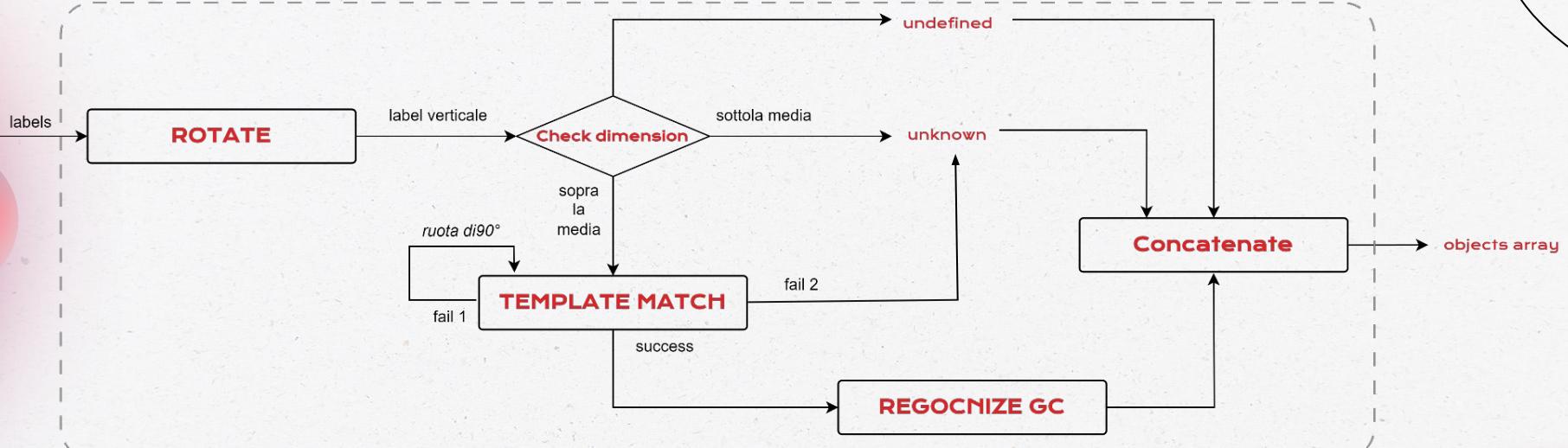
Escludi



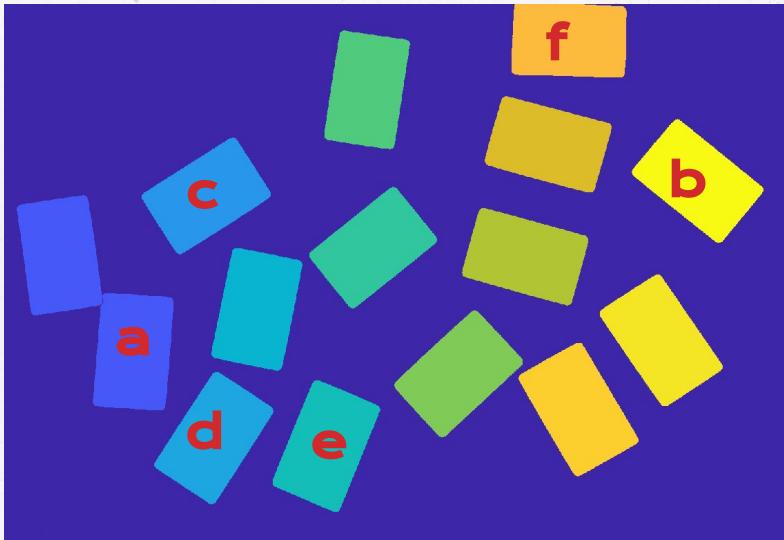
06

OBJECT RECOGNITION

OBJECT RECOGNITION



INPUT E OUTPUT

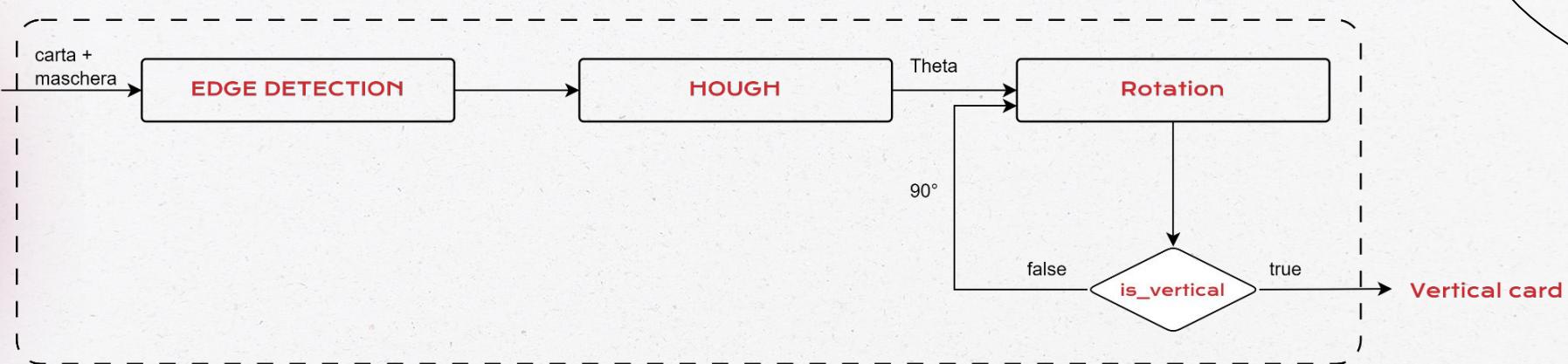


Labels dell' immagine n° 21 in input a object recognition

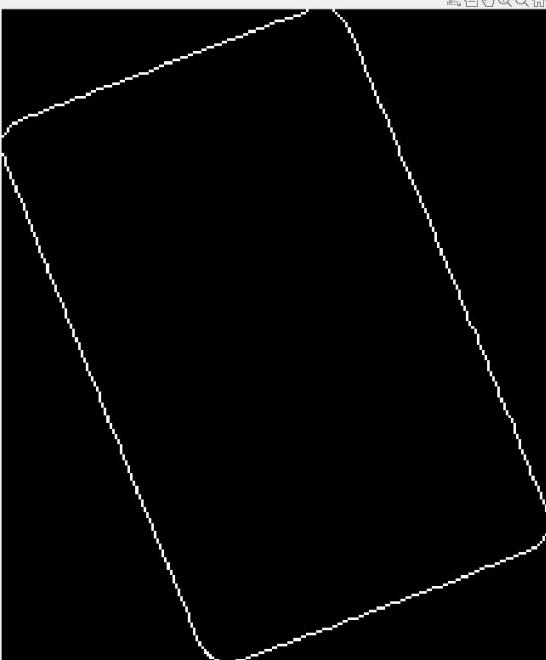


Alcune delle carte estratte da object recognition

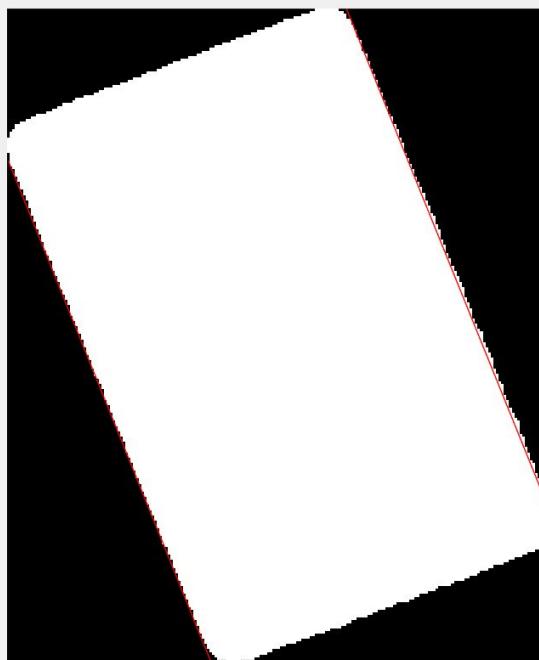
ROTATE



FUNZIONAMENTO



Plot degli edge individuati con Prewitt

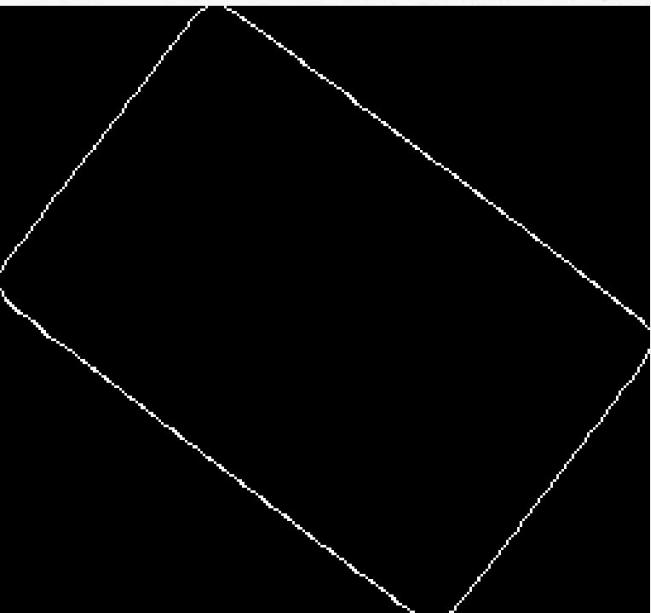


Plot delle rette individuate da hough sulla maschera

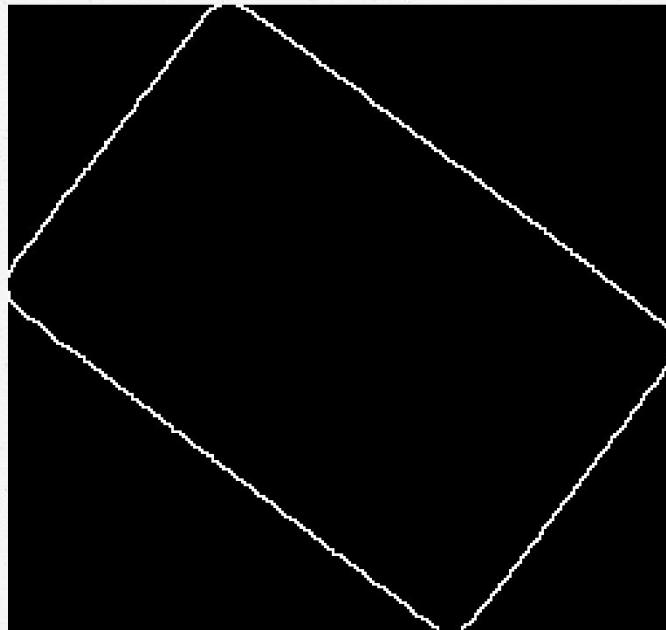
Come prima cosa viene effettuato un **edge detection**

L' algoritmo di Hough lavora sull' img degli edge

ESEMPI DI EDGE

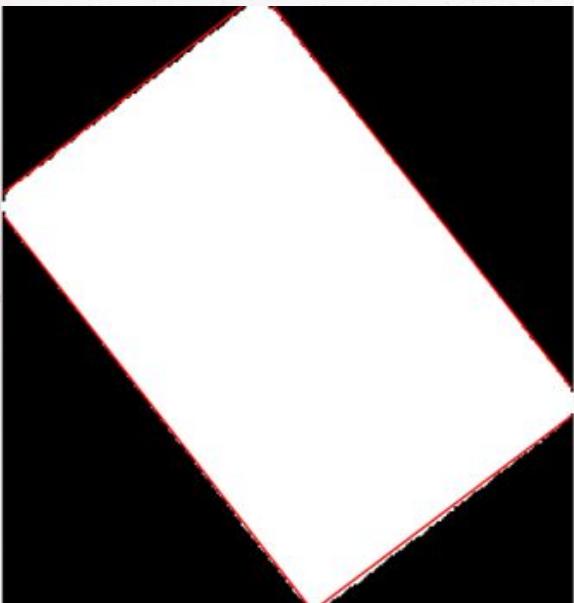


Plot degli edge individuati con Sobel

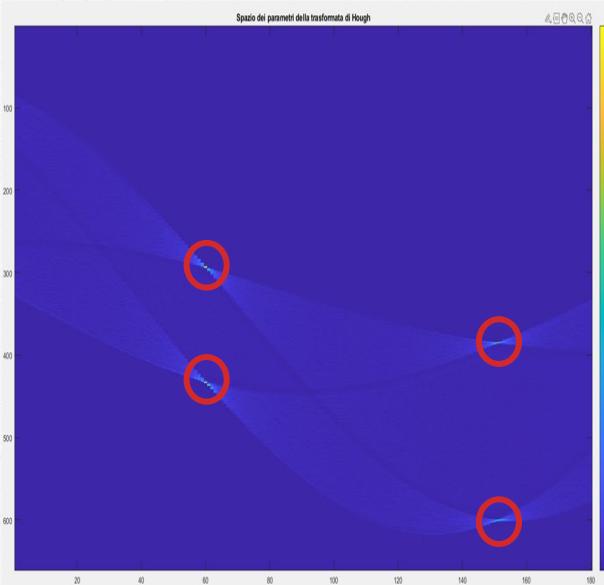


Plot degli edge individuati con Roberts

FUNZIONAMENTO



Plot delle rette individuate da Hough sulla maschera

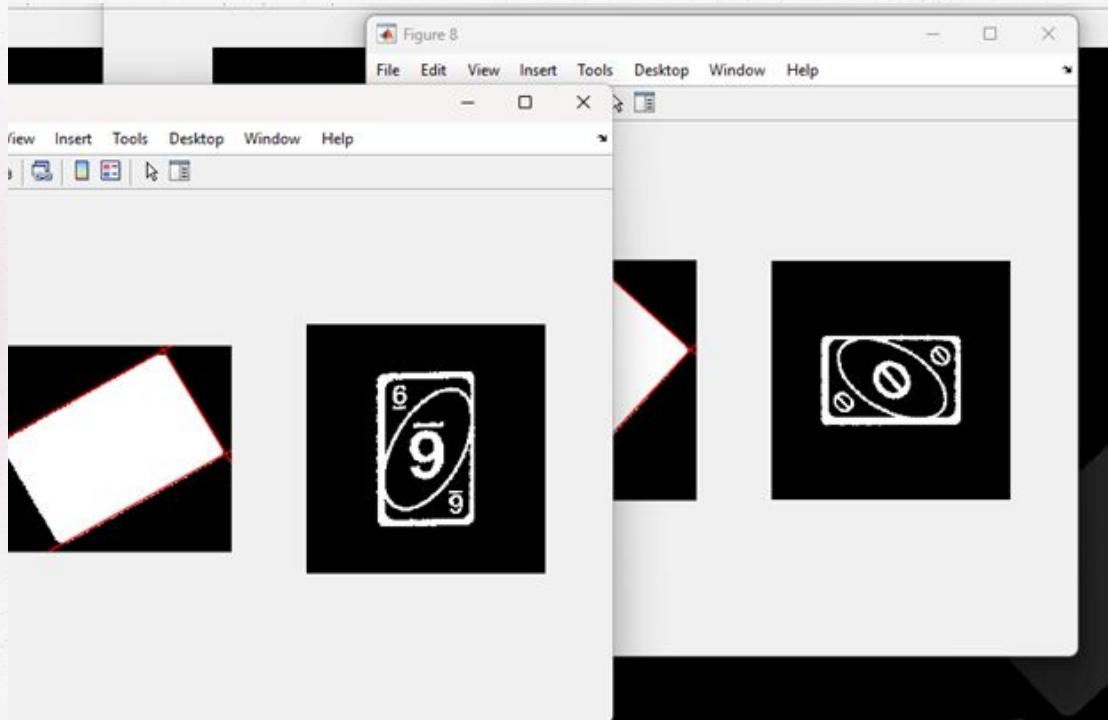


Plot grafico dei parametri di Hough

Tramite Hough è possibile estrarre le **4 rette** corrispondenti ai lati della carta

E il loro **coefficiente angolare**

IMPRECISIONI



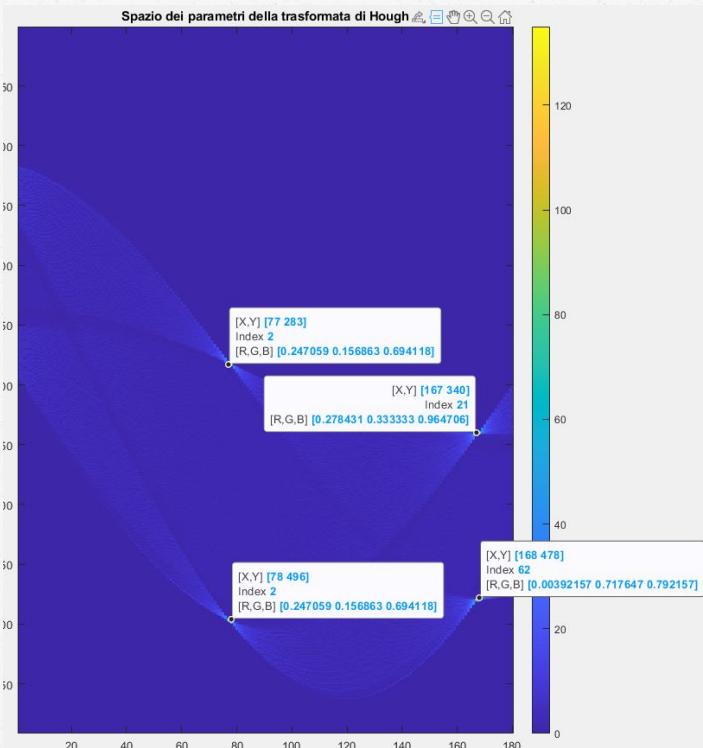
Plot dell'individuazione di sole 3 rette

Plot di una possibile carta girata in orizzontale

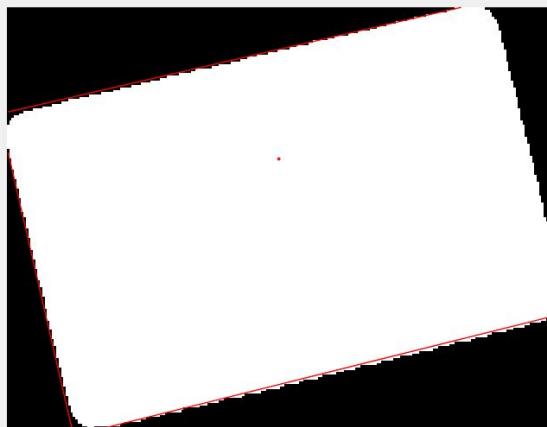
Risolte prelevando sempre uno dei due lati maggiori

Ruotando di 90 gradi se la carta è in orizzontale

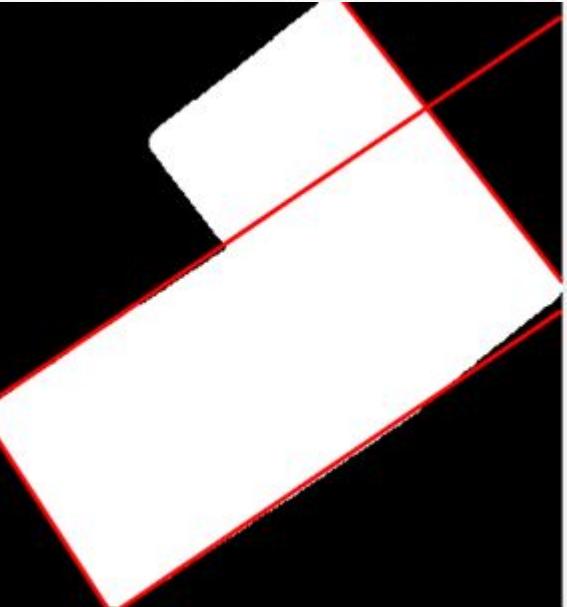
ALTRÉ IMMAGINI



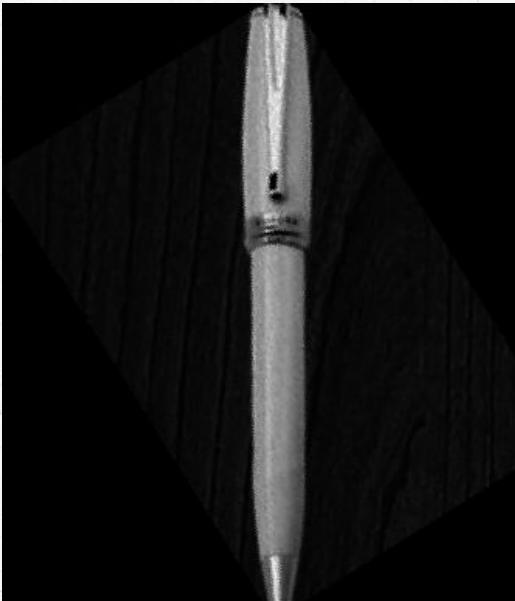
Plot mostrante l' imprecisione dell' algoritmo



ALTRÉ IMMAGINI



Plot dell'individuazione di rette nei gruppi di carte



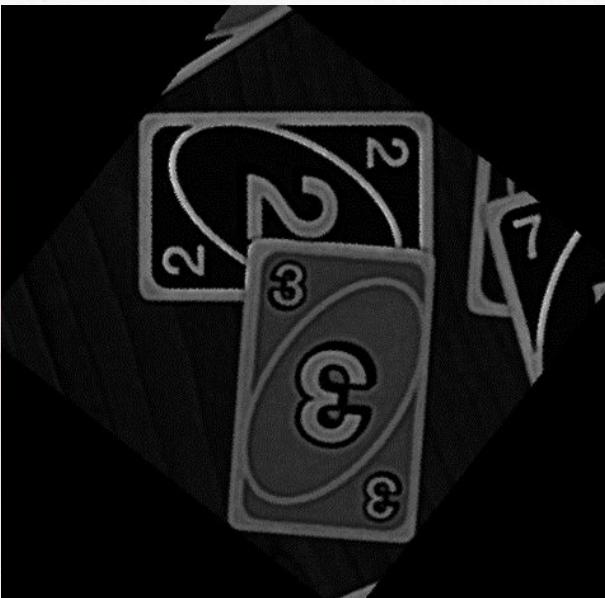
Esempio di rotazione applicata ad una penna

Abbiamo testato che il nostro algoritmo riesce ad individuare le **rette** anche nei **gruppi di carte**

Permettendo la loro rotazione

Qualsiasi oggetto **estraneo** nella scena, inoltre viene ruotato senza problemi

ALTRÉ IMMAGINI



Plot di un gruppo di carte girato rispetto alla carta 3

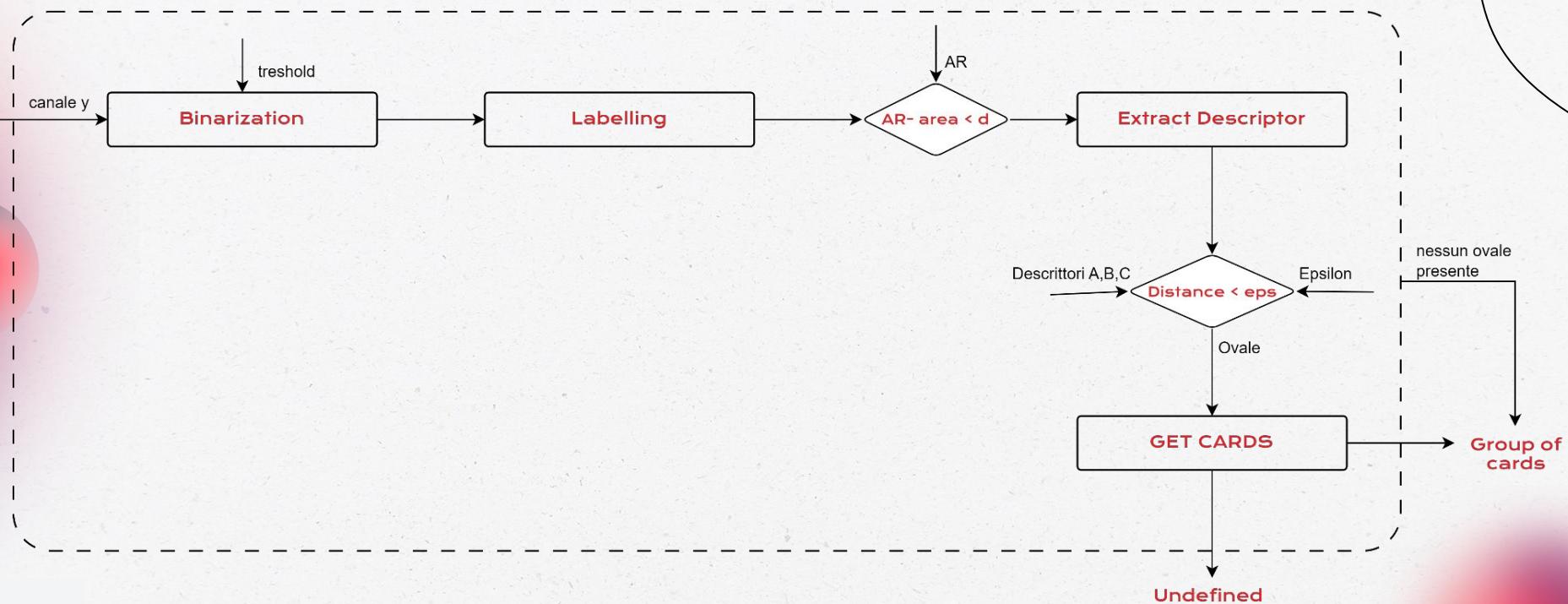


Plot del gruppo di carte girato per fare Template Matching

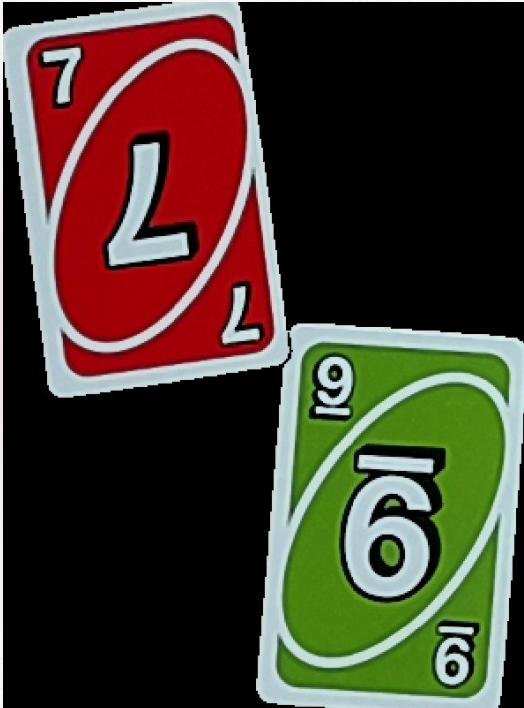
Alla fine le carte e i gruppi di carte possono essere passati al classificatore poiché hanno almeno un **simbolo** in verticale

Il riconoscimento verrà effettuato tramite **template verticali**

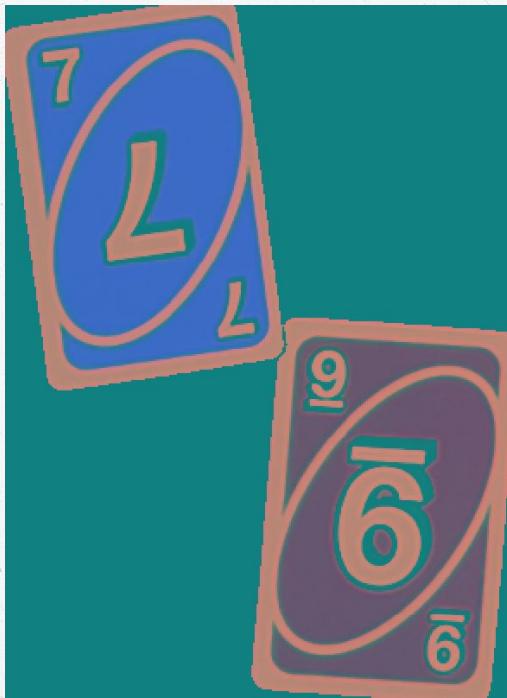
RECOGNIZE GC



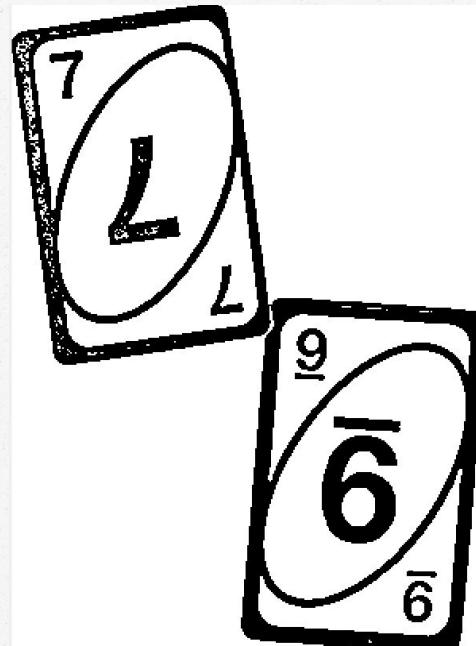
LABELLING



Plot gruppo di carte RGB

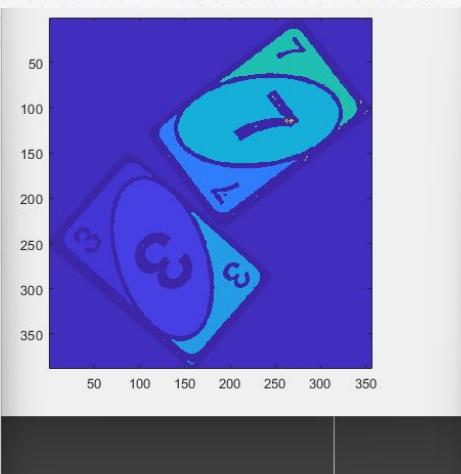
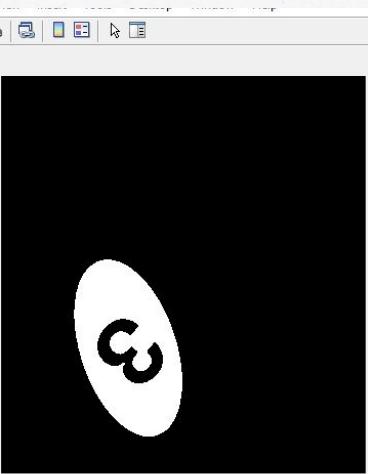


Plot gruppo di carte YCBCR



Plot negativo della sogliatura sul canale Y con $T = 0.7059$

Scrematura ovali



Plot dell' estrazione degli ovali tramite l' area media

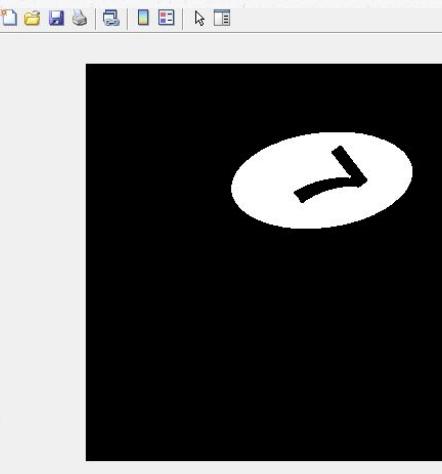


Figura 1

Un primo partizionamento delle labels avviene grazie all' area media:

I AR - area I <= delta, dove **AR 11000 px e delta = 2100 px**

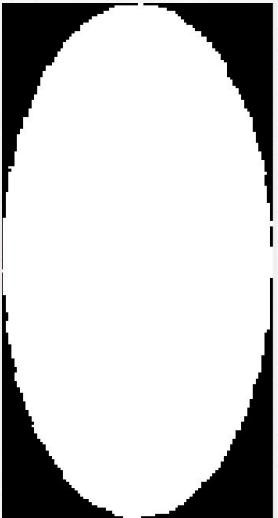
Un' operazione **close** e una **rotate** restituiscono l' ovale come in **figura 1**

[2p/A, 2p, A/back] - [A, B, C] <= [eps1, eps2, eps3]

$A = 3.13e-2, B = 4e+2, C = 3.45e+2$

$\text{eps1} = 2e-3, \text{eps2} = \text{eps3} = 3e+1,$

Alternative Considerate signature



L'idea era quella di **campionare i pixel** di bordo della labels e calcolare la loro **distanza dal centro**

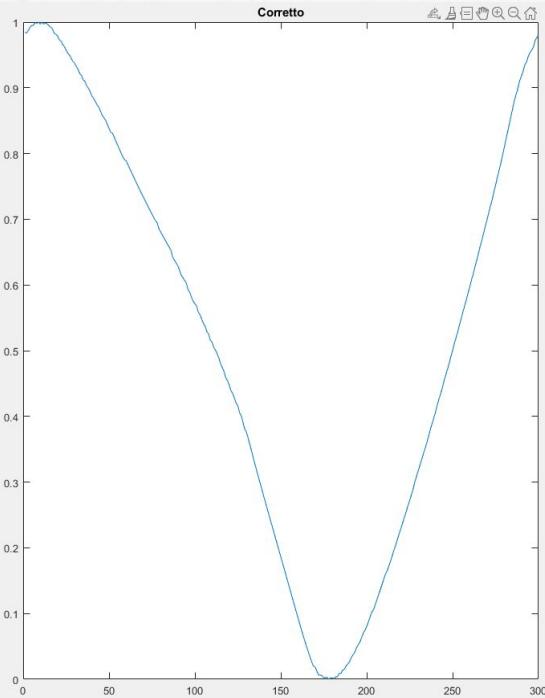
L'idea era quella di **campionare i pixel** di bordo della labels e calcolare la loro **distanza dal centroide**

Campioni = 400

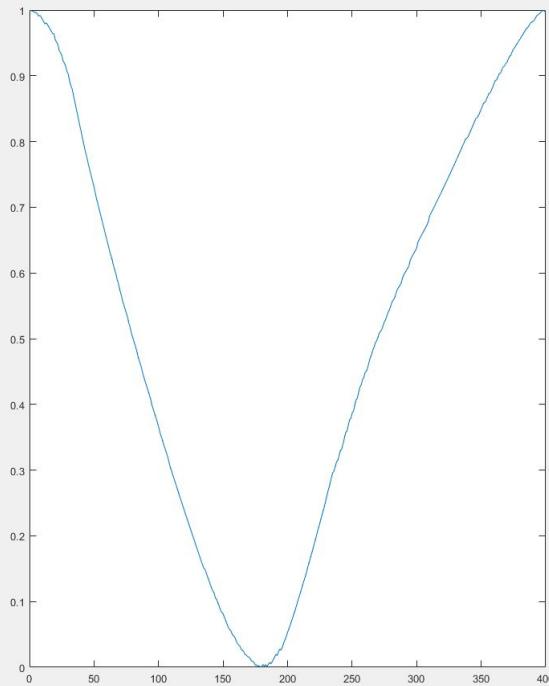
Passo = $2p / \text{Campioni}$

Campioni e passi differenti portavano alla stessa **distanza euclidea** errata

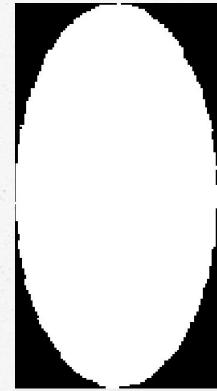
Alternative Considerate signature



Plot signature ovale A di riferimento



Plot signature ovale B

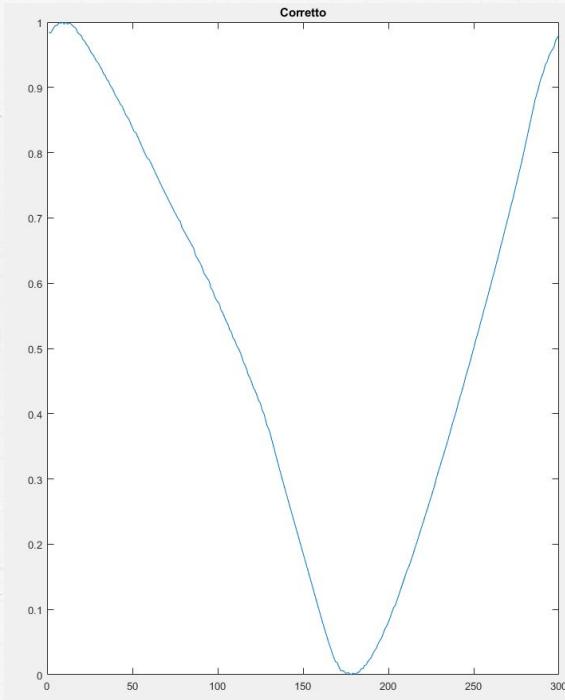


Plot ovale A,

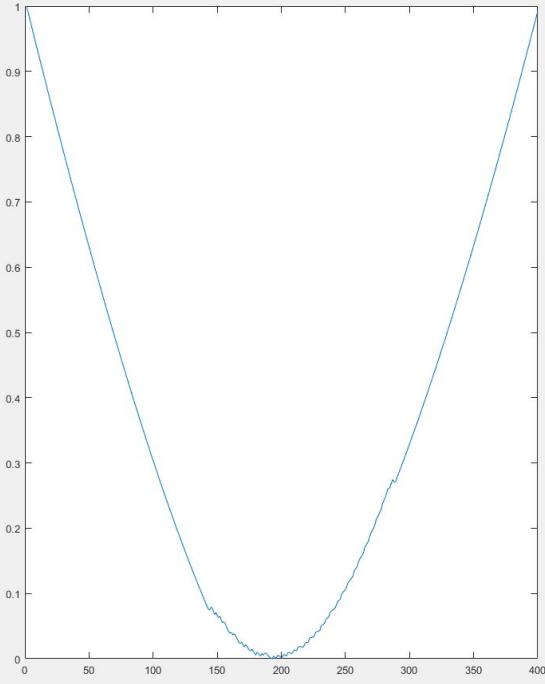


Plot ovale B

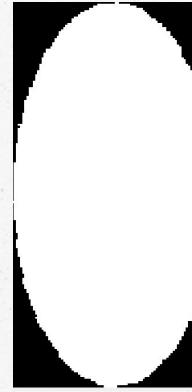
Alternative Considerate signature



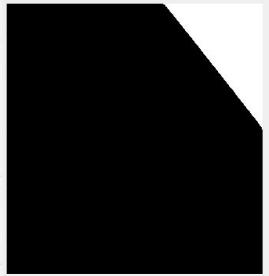
Plot signature ovale A di riferimento



Plot signature label B

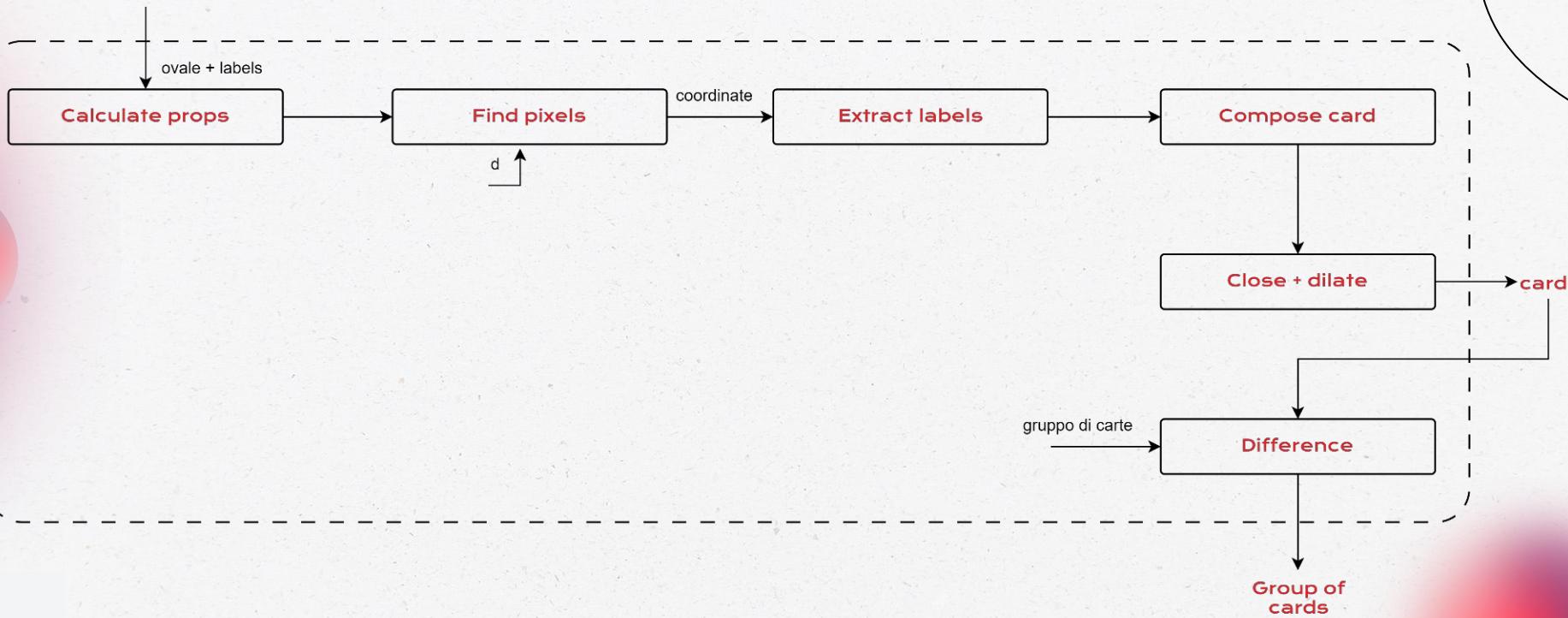


Plot ovale A

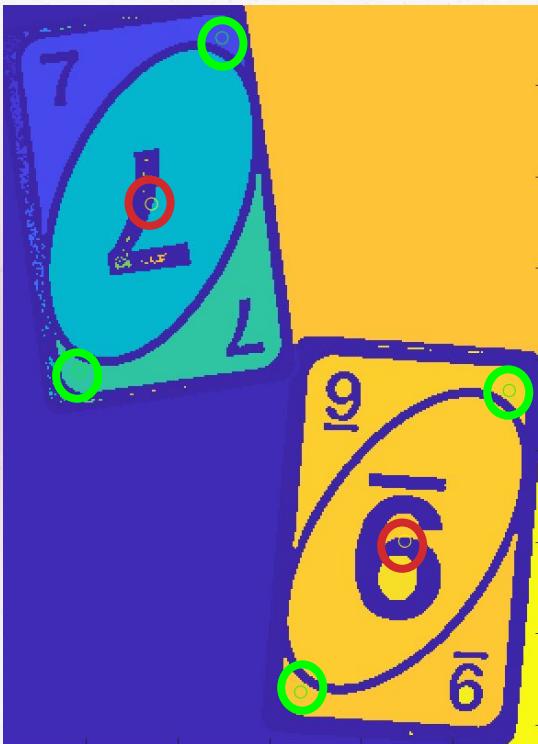


Plot ovale B

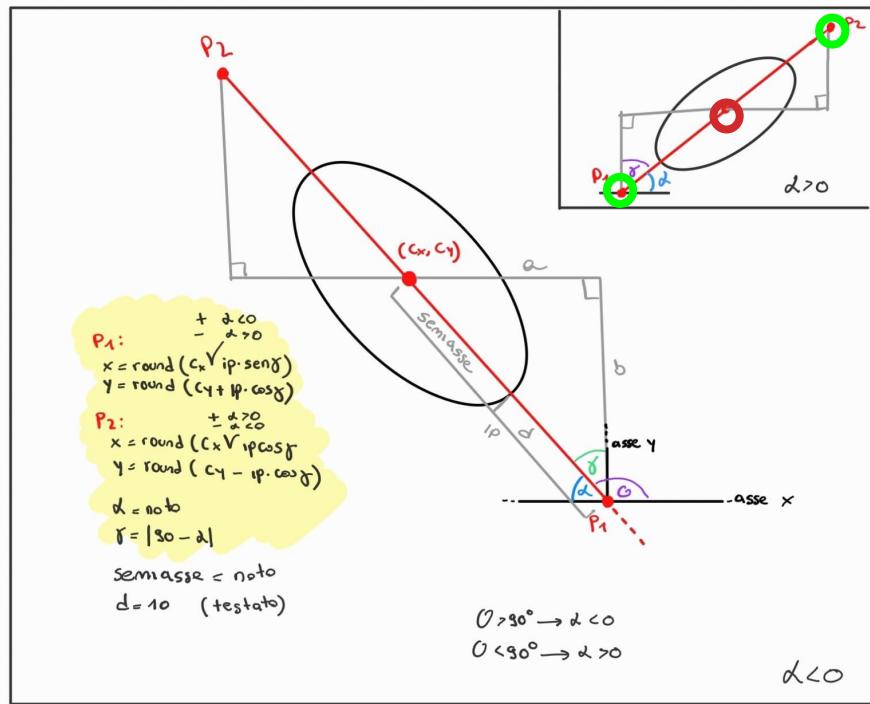
MICRO MODULI - GET CARDS



FIND PIXEL



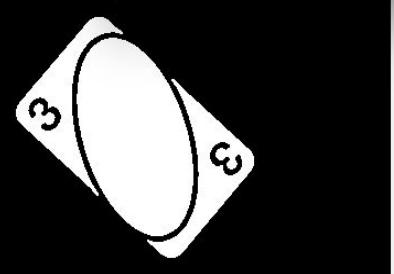
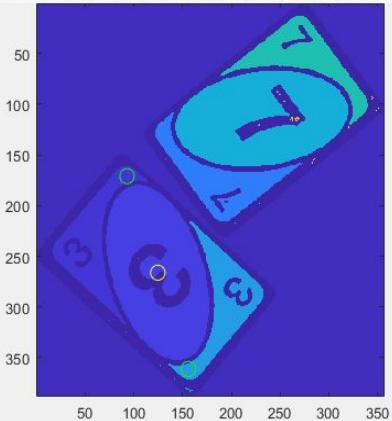
Plot centroide e punti P_1 e P_2



Schema calcoli trigonometrici

Tramite test,
abbiamo
constatato
che d vale in
media 10

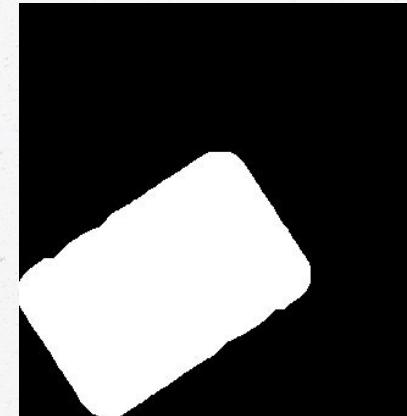
FIND PIXEL



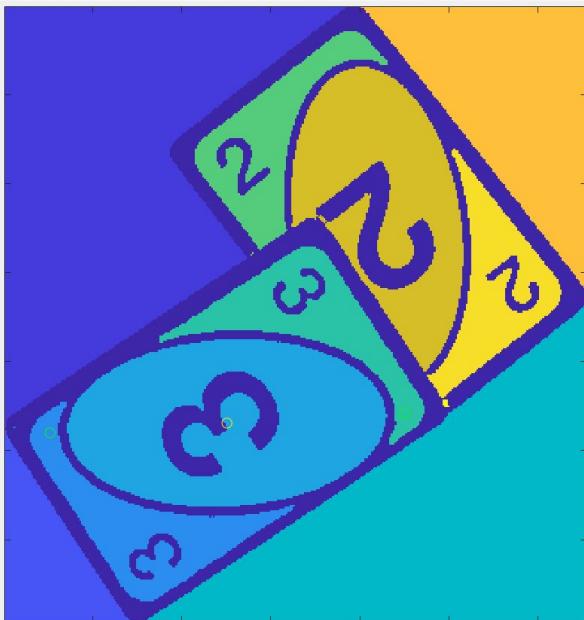
Plot estrazione carta



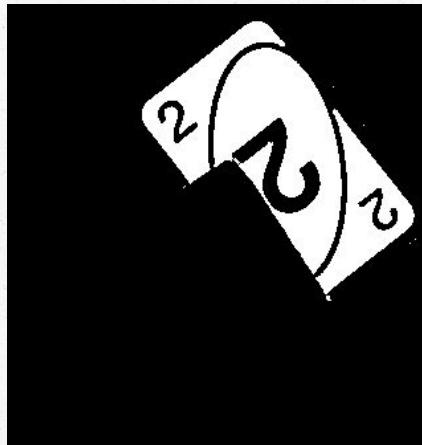
Plot estrazione close + dilate
carta estratta e maschera RGB



FIND PIXEL



Plot labels gruppo di carte



Plot maschera rimanente



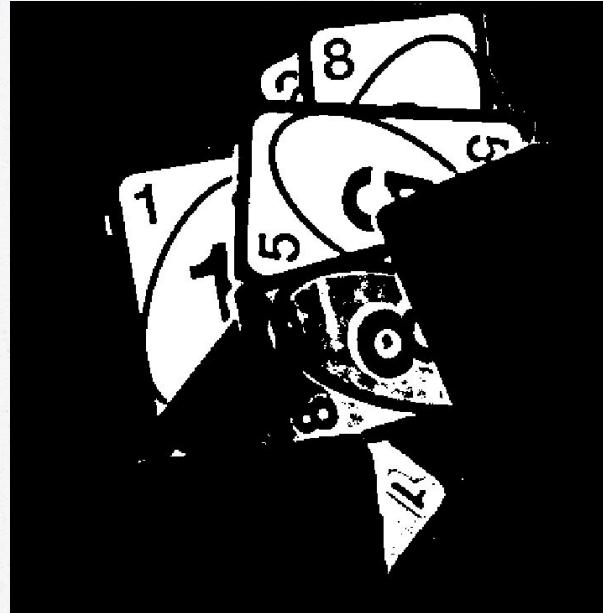
Plot maschera RGB

Per le nostre assunzioni, la carta 2 **non è riconoscibile** di conseguenza verrà classificata come **gruppo di carte**

ALTRI ESEMPI



Plot immagine di input



Plot gruppo di carte non riconosciuto



Plot carte riconosciute

Abbiamo un'idea di come separare ulteriormente le carte, combinando **template matching** e **region growing**

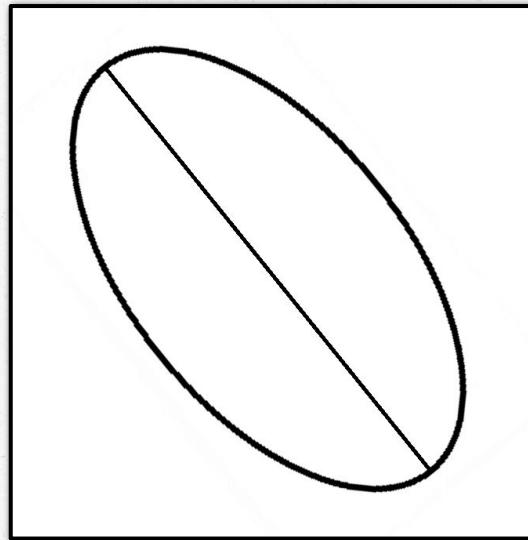
Non è stato però un obiettivo primario

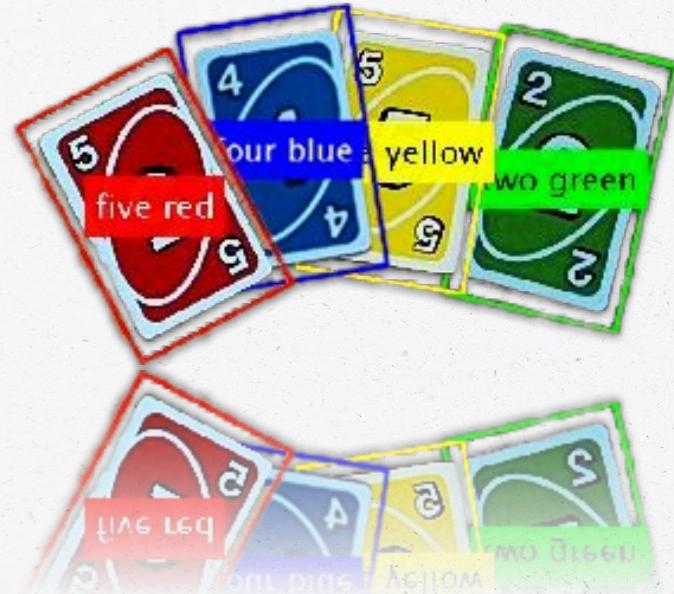
Alternative Considerate

algoritmo di Bresenham

Abbiamo pensato di usare questo algoritmo per congiungere i due punti estremi della Bounding Box dell'ovale, in modo da ottenere l'asse maggiore.

Scartato a seguito di individuazione di algoritmo più efficiente.

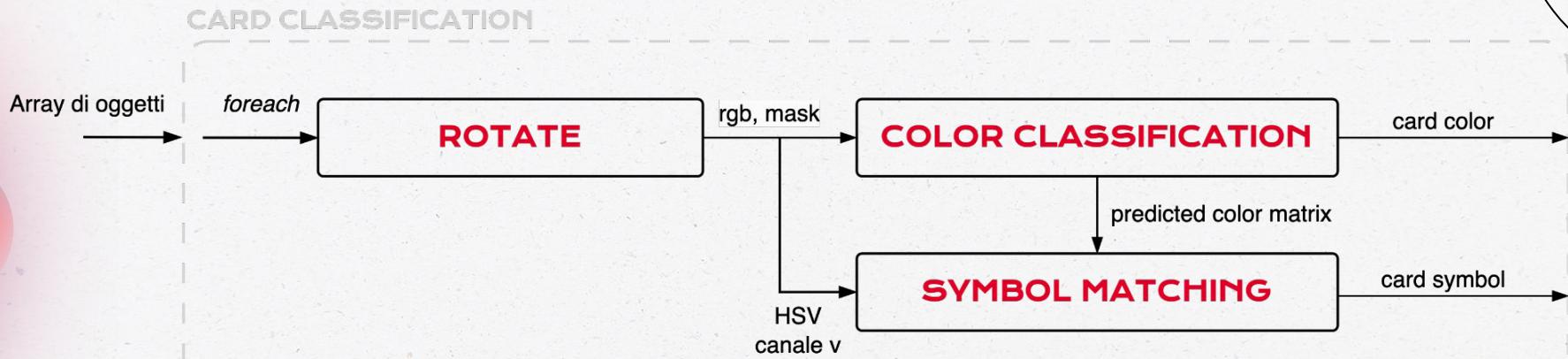




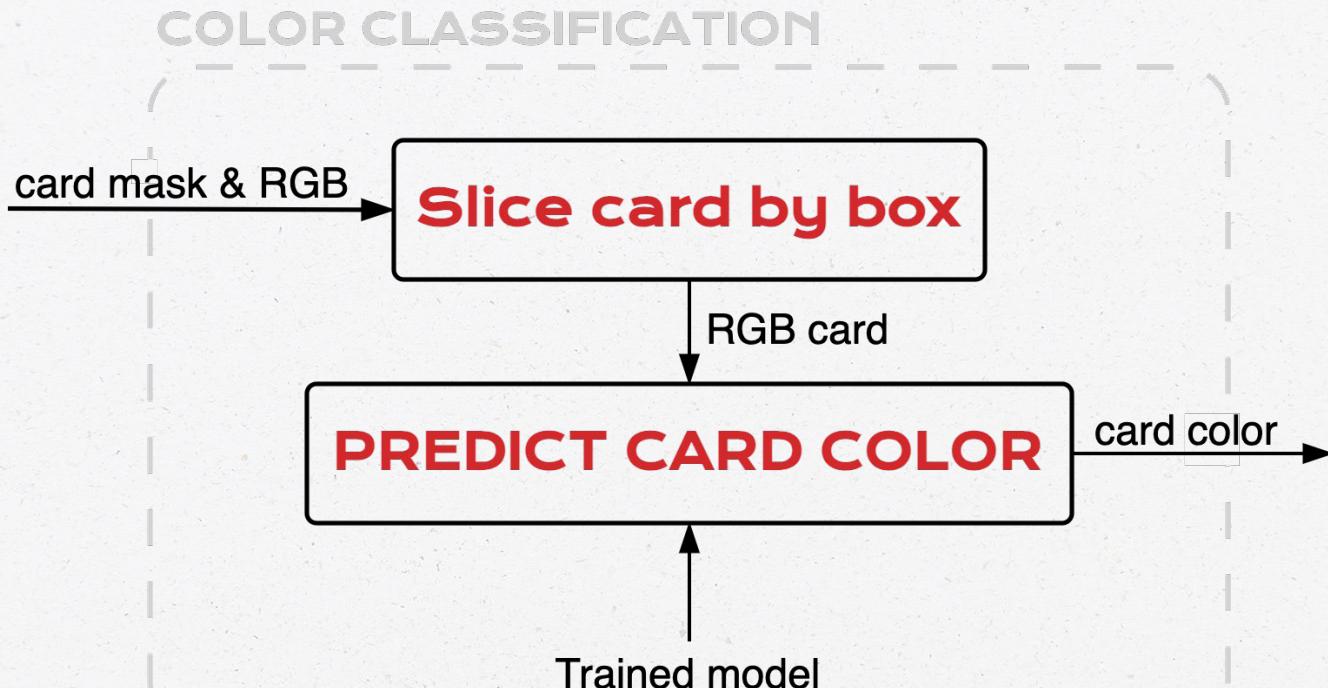
07

CARD CLASSIFICATION

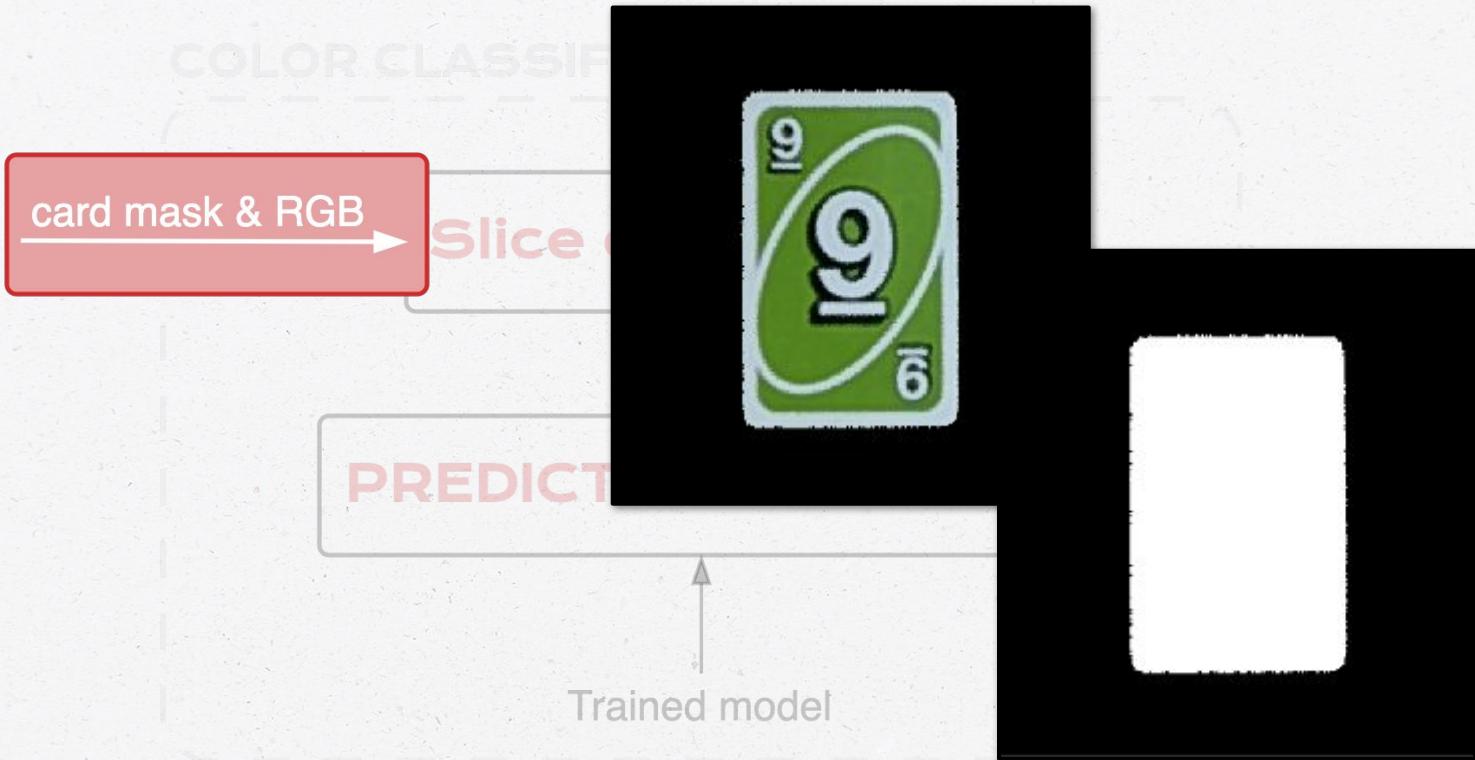
CLASSIFICATION PIPELINE



COLOR CLASSIFICATION PIPELINE



COLOR CLASSIFICATION PIPELINE



COLOR CLASSIFICATION PIPELINE



CLASSIFICATION

Slice card by box

RGB card

PREDICT CARD COLOR

Trained model

card color

COLOR CLASSIFICATION PIPELINE



COLOR
CLASSIFICATION

Slice card by box

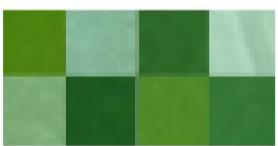
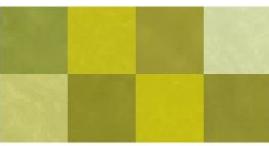
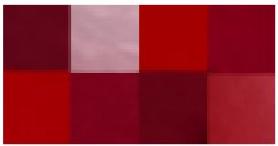
RGB card

PREDICT CARD COLOR

card color →

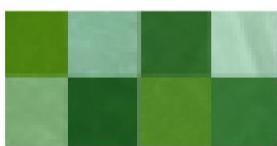
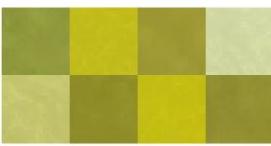
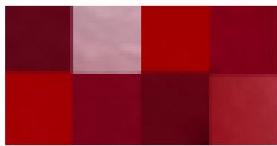
Trained model

MODEL TRAINING



Il dataset di training contiene una gamma di pixel per ogni colore: Rosso, Giallo, Verde, Blu, Nero e Altro.

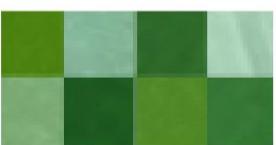
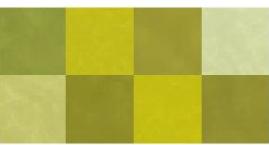
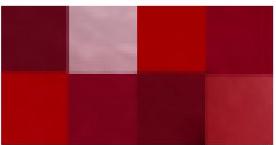
MODEL TRAINING



Il dataset di training contiene una gamma di pixel per ogni colore: Rosso, Giallo, Verde, Blu, Nero e Altro.

Attualmente il dataset è piuttosto piccolo: si tratta di 80x40px per colore.

MODEL TRAINING



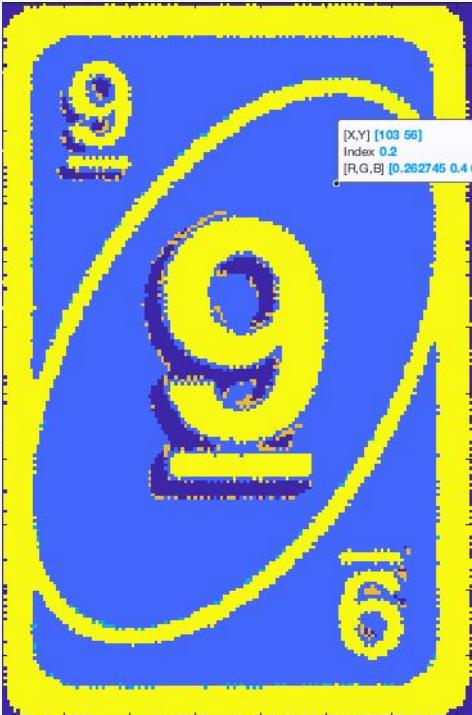
Il dataset di training contiene una gamma di pixel per ogni colore: Rosso, Giallo, Verde, Blu, Nero e Altro.

Attualmente il dataset è piuttosto piccolo: si tratta di 80x40px per colore.

Ogni colore ha associato un valore tra 0 e 1

Red: 0.8	Yellow: 0.4
Green: 0.2	Blue: 0.6
Black: 0	Altro: 1

MODEL TRAINING

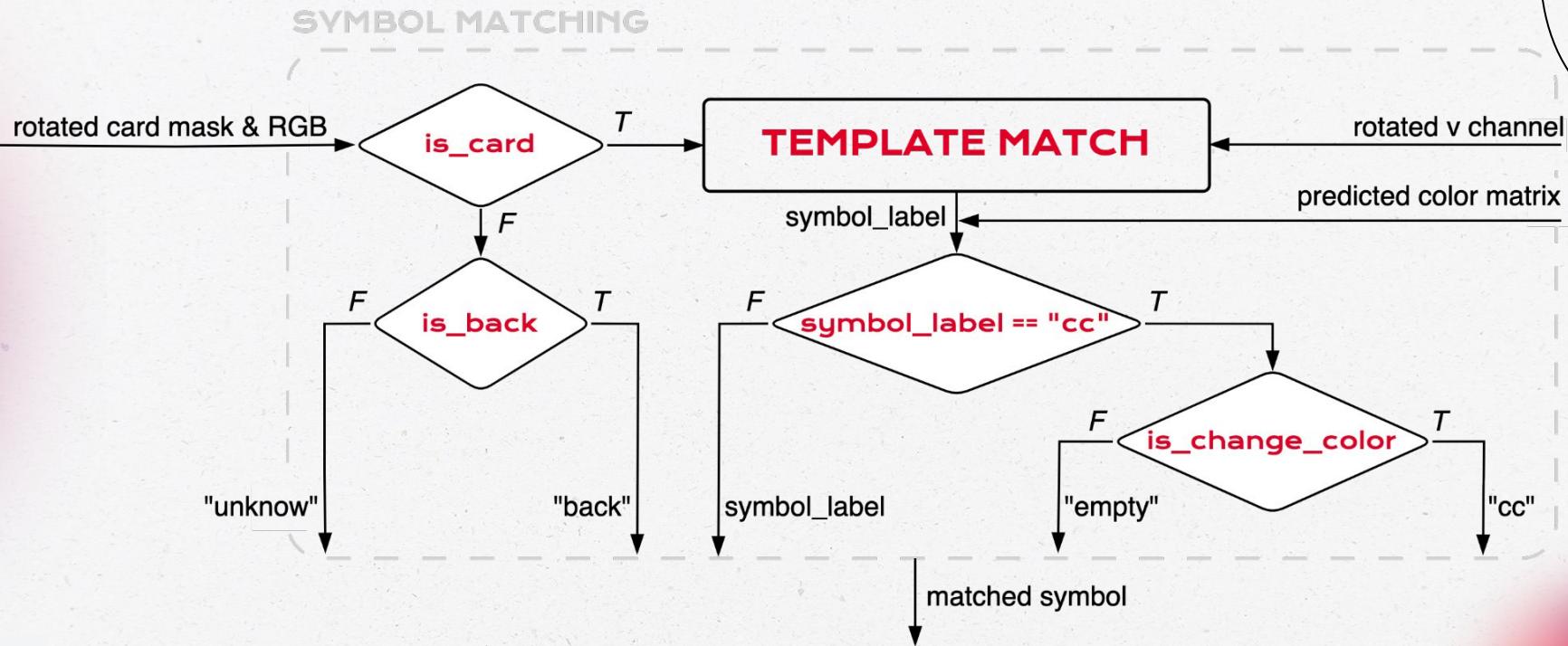


0.2 corrisponde al verde nella nostra LUT

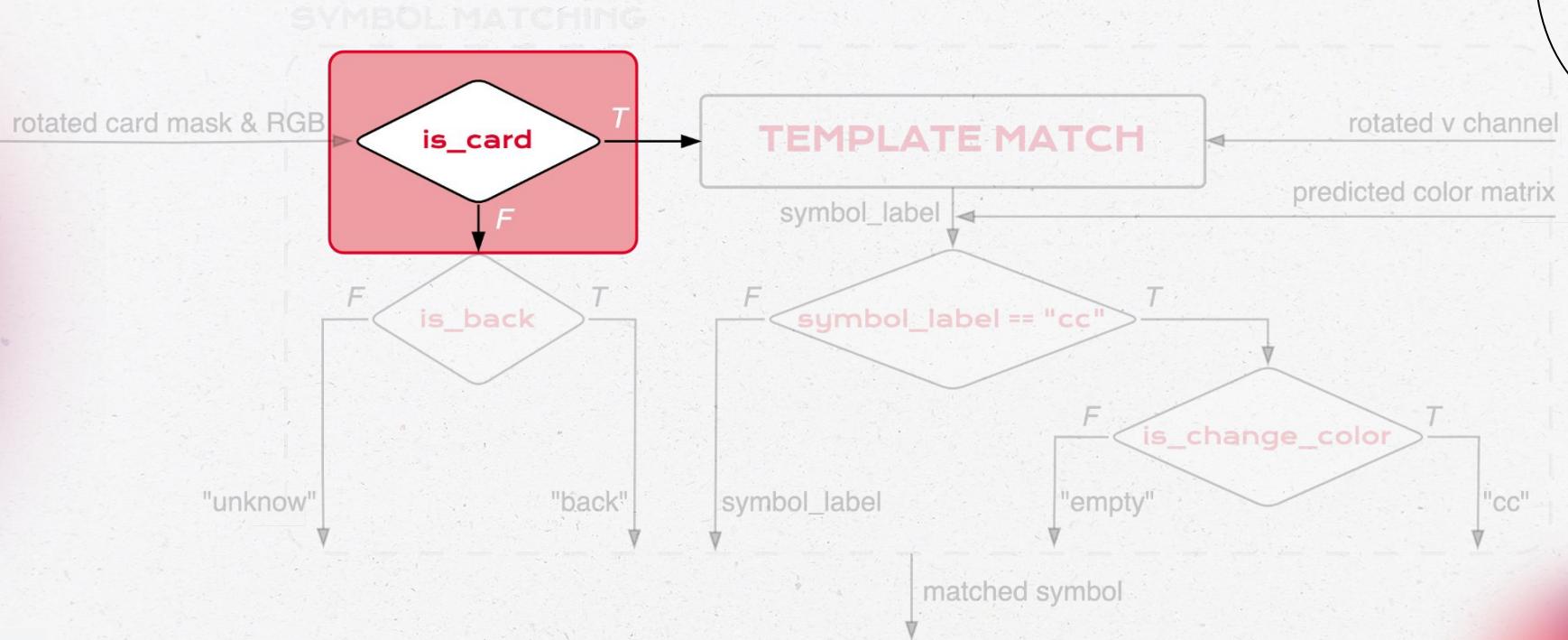
Questo ci consente di calcolare la moda della classificazione del colore, in modo da ottenere il valore tra 0 e 1 più frequente e di conseguenza il colore corrispondente.



SYMBOL MATCHING PIPELINE



SYMBOL MATCHING PIPELINE



Is a card



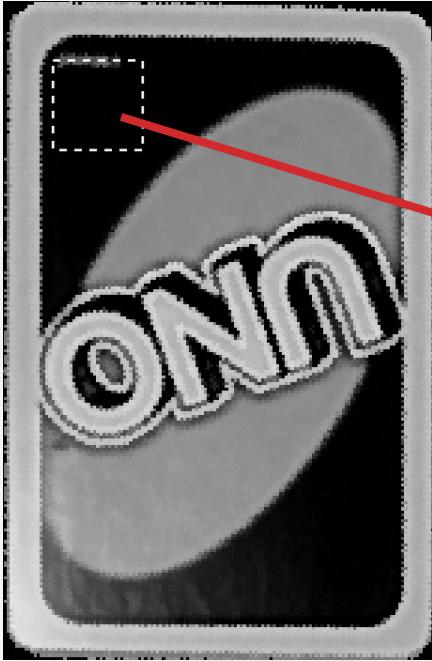
Secondo le nostre assunzioni, una carta è definibile come tale e quindi classificabile se ha il simbolo presente e visibile in alto a sinistra.

Le **carte girate** non hanno nessun simbolo, quindi avranno il **rettangolo** completamente **nero**

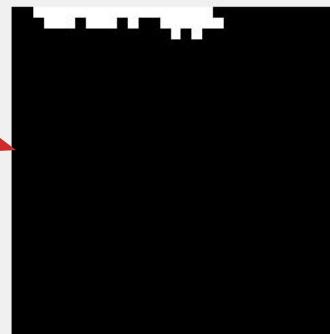
$$R * 100 > 95$$

R = pixel neri / pixel quadrato
Pixel quadrato = 30 px

Is a card



Plot canale V carta girata

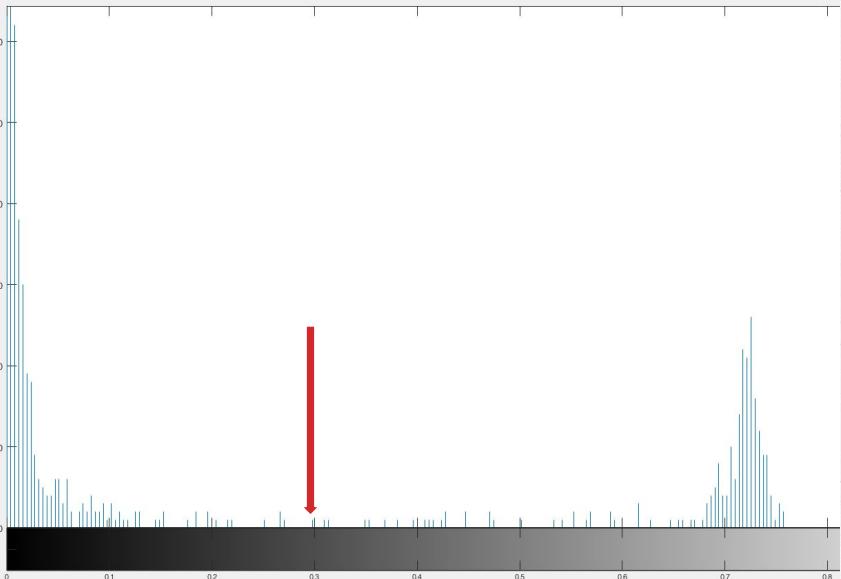


Plot quadrato estratto e binarizzato

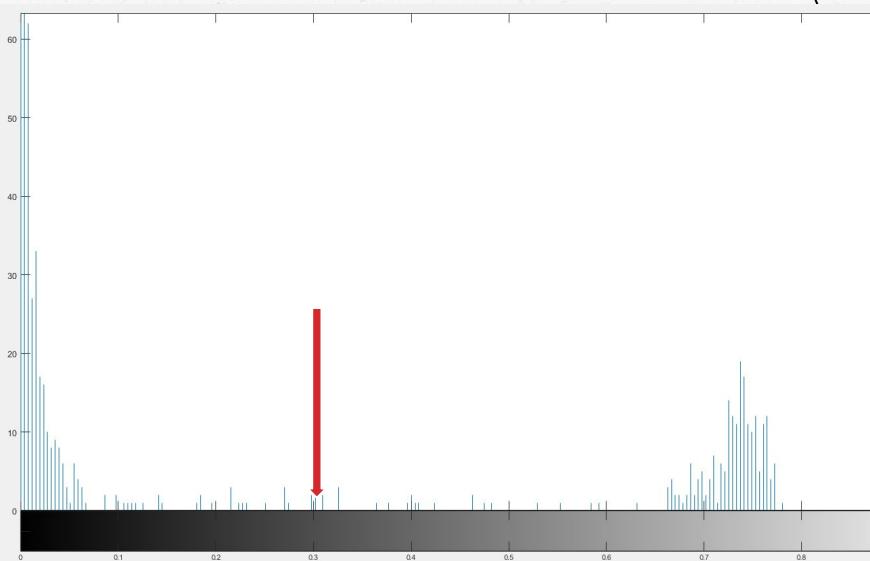
Si utilizza il canale **V** di HSV e si binarizza con **$t >= 0.32$**

Ricavata in seguito a numerosi test e studi dell'istogramma

Is a card

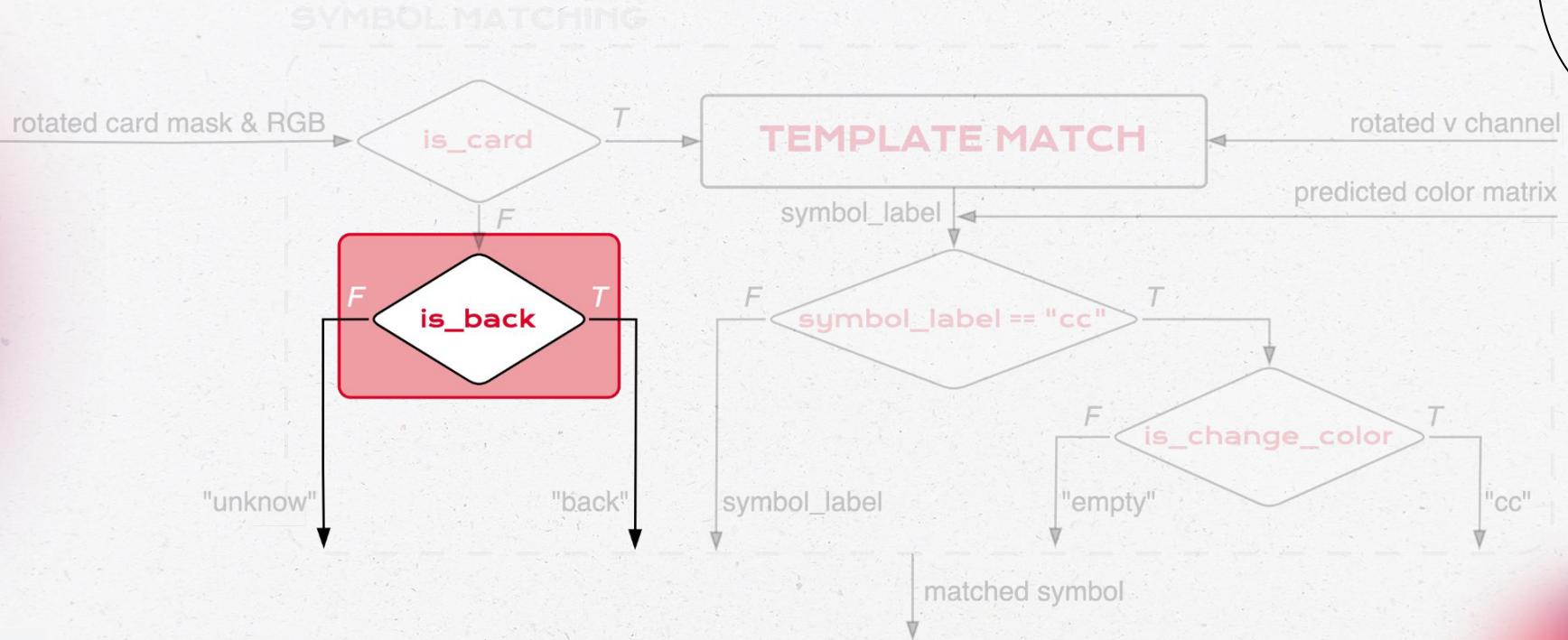


Plot 1 istogramma canale V quadrato di una carta girata

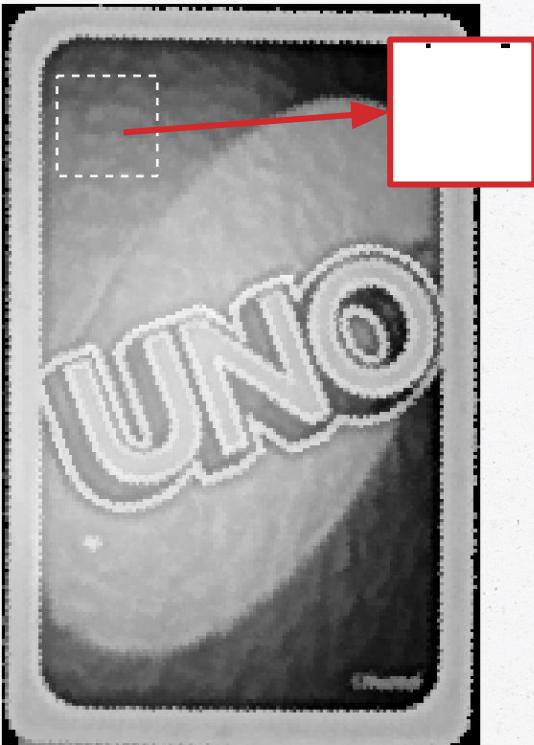


Plot 2 istogramma canale V quadrato di una carta girata

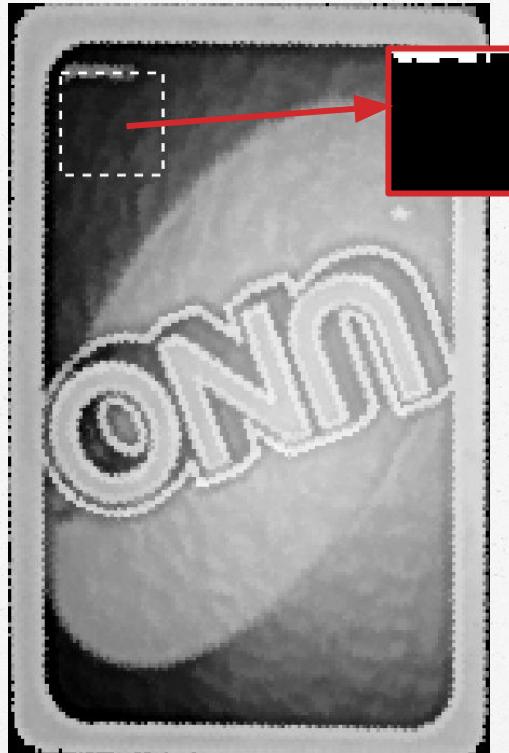
SYMBOL MATCHING PIPELINE



Is back



Plot canale V carta soggetta a forte illuminazione + square



Plot canale V carta soggetta a forte illuminazione girata + square

Il controllo del quadrato viene effettuato anche su **carta girata di 180°**, per ovviare ai problemi di luce

La limitazione è nelle carte con **luce non omogenea**

Is back



Plot carta di output dal classificatore del colore

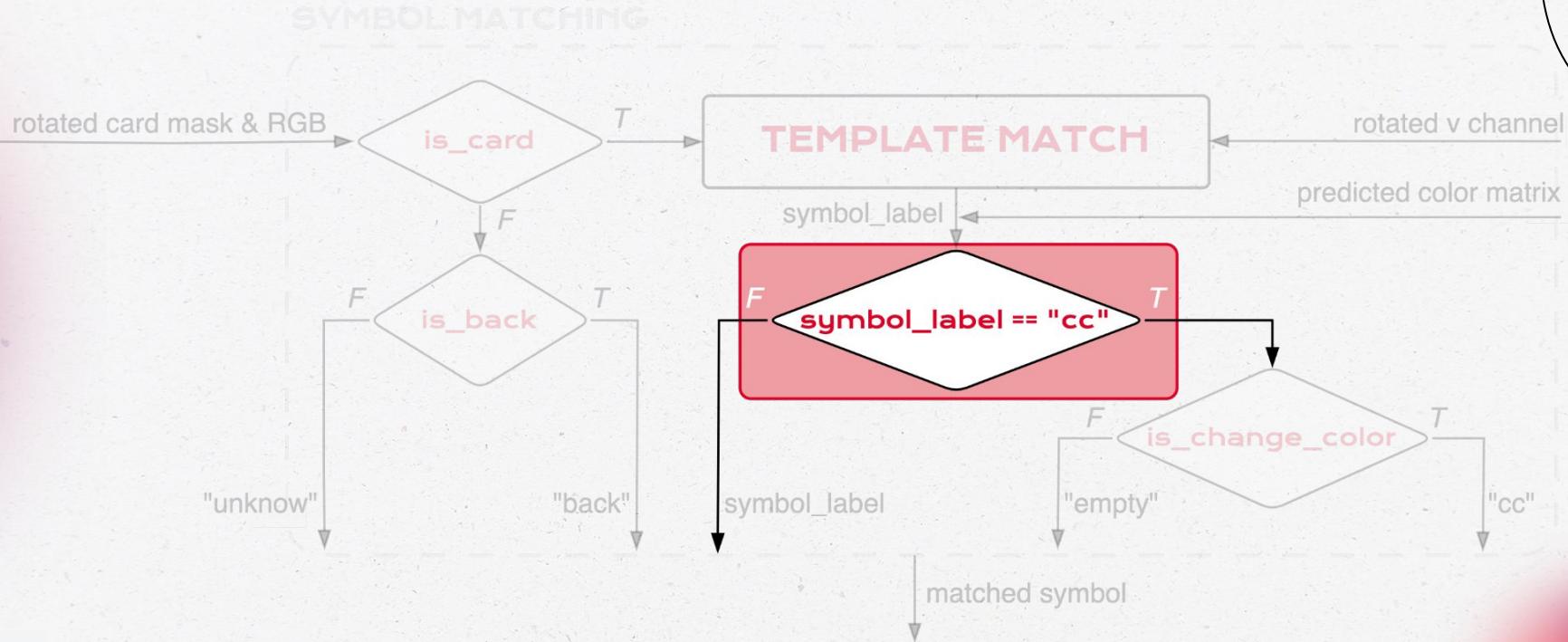


Plot binarizzazione rispetto ai pixel rossi

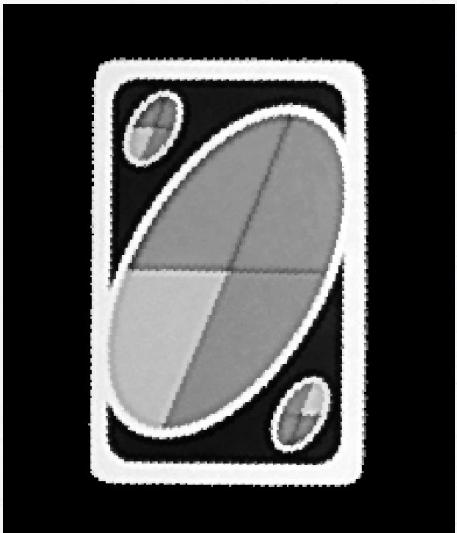
Si classificano i pixel e si estraggono solo quelli rossi

A / back * 100 > 50

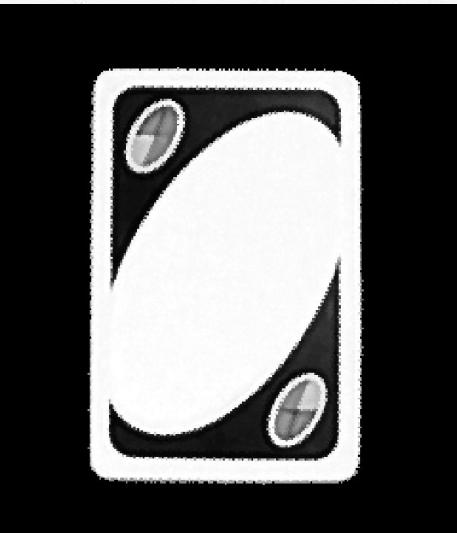
SYMBOL MATCHING PIPELINE



Cambio colore o carta nulla?



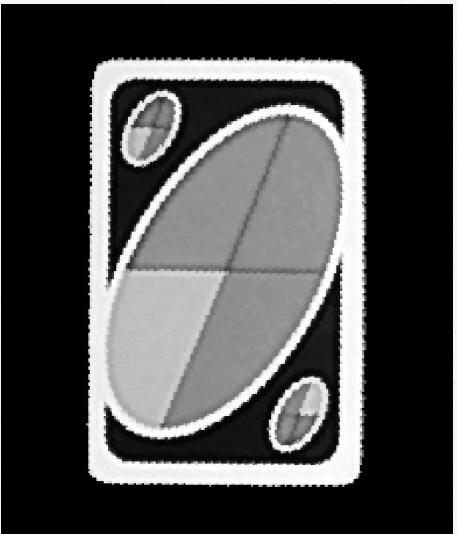
29.16%



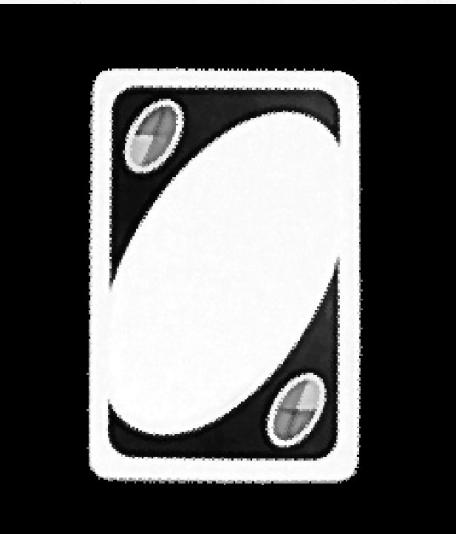
69.95%

I simboli delle due carte sono gli stessi, dunque il template matching restituisce “cambio colore” in entrambi i casi

Cambio colore o carta nulla?



29.16%

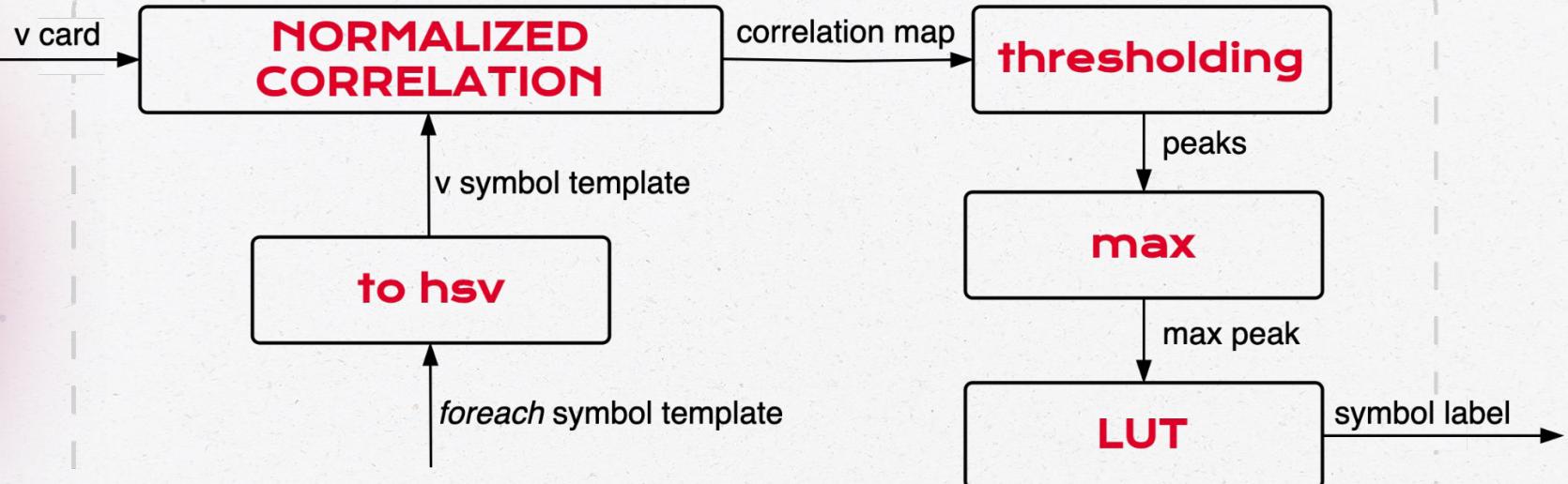


69.95%

I simboli delle due carte sono gli stessi, dunque il template matching restituisce “cambio colore” in entrambi i casi

Separiamo una carta vuota da una carta colore in base alla percentuale di pixel bianchi rispetto al resto

TEMPLATE MATCH



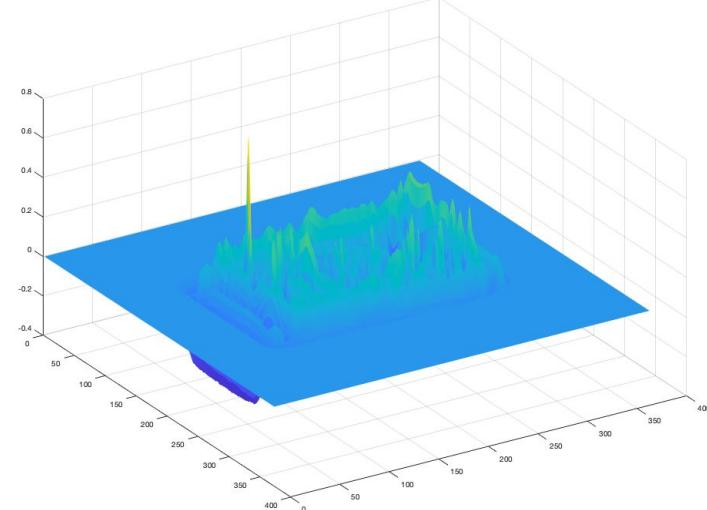
Template == Card Symbol



Carta con simbolo da riconoscere



Template del simbolo 2



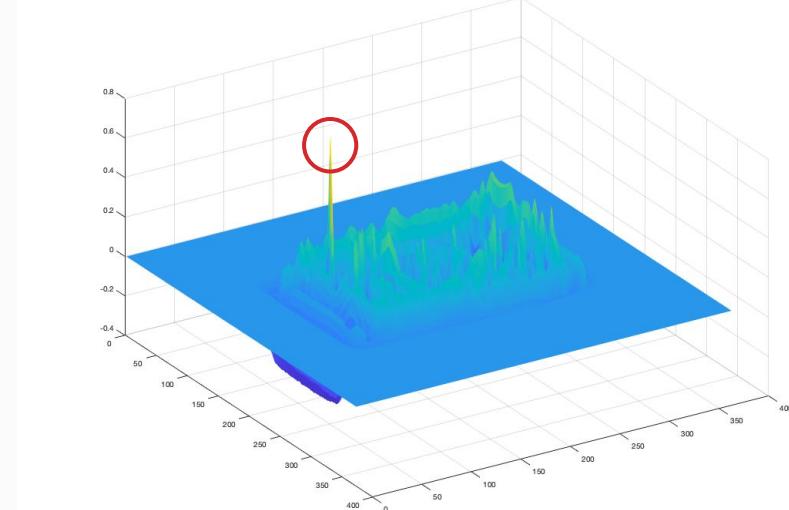
Mappa di correlazione 3d tra la carta e il template

Template == Card Symbol

0.7638

Il valore del picco nella
mappa di correlazione

Carta con simbolo da riconoscere



Mappa di correlazione 3d tra la carta e il template

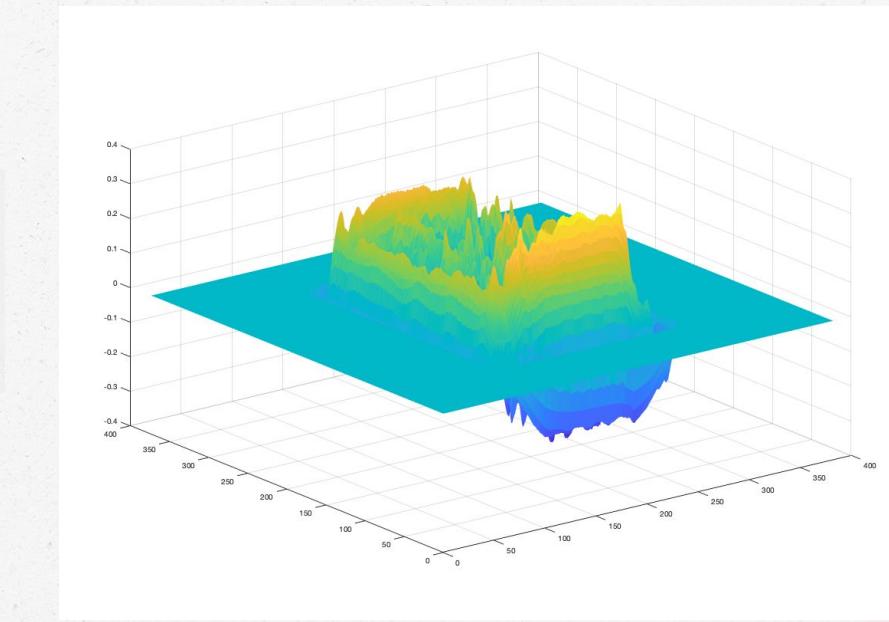
Template != Card Symbol



Carta con simbolo da riconoscere



Template del simbolo +4



Mappa di correlazione 3d tra la carta e il template

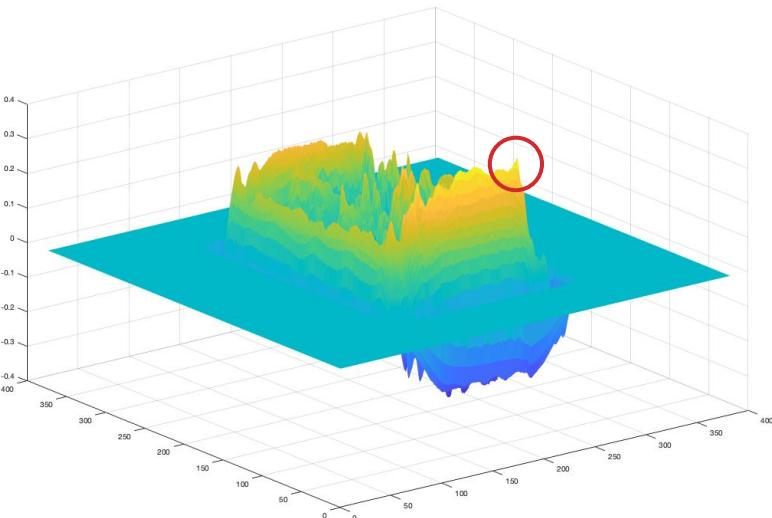
Template != Card Symbol

0.3678

Il valore del picco nella
mappa di correlazione

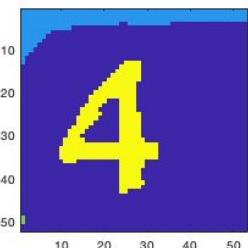
Sicuramente questo simbolo
verrà scartato dal thresholding

Carta con simbolo da riconoscere

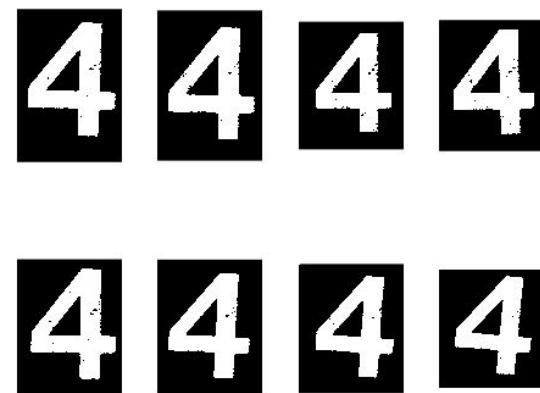


Mappa di correlazione 3d tra la carta e il template

Alternative Considerate array difference



Sezione in alto a sinistra della carta



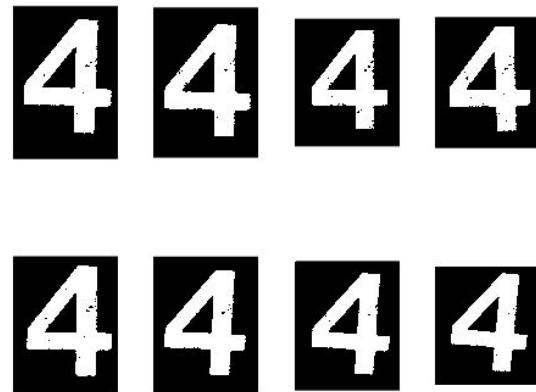
Array di simboli con i quali fare la differenza

Alternative Considerate array difference

Per ogni possibile simbolo, vi era un array di templates



Sezione in alto a sinistra della carta



Array di simboli con i quali fare la differenza

Alternative Considerate array difference

Per ogni possibile simbolo, vi era un array di templates

Eseguivamo la differenza tra ogni simbolo nell'array e la sezione in alto a sinistra nella carta, per ogni possibile simbolo



Sezione in alto a sinistra della carta



Differenza tra simboli diversi



Alternative Considerate array difference

Per ogni possibile simbolo, vi era un array di templates

Eseguivamo la differenza tra ogni simbolo nell'array e la sezione in alto a sinistra nella carta, per ogni possibile simbolo

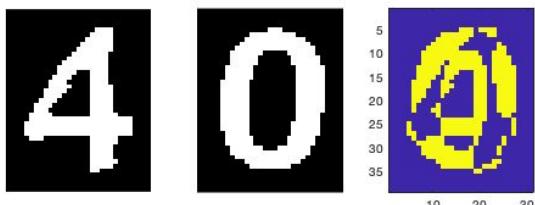
L'array di differenze ottenuto veniva comparato con tutti gli altri array di differenze per altri simboli. Quello con distanza minima corrispondeva al simbolo esatto

Sezione in alto a sinistra della carta



Differenza tra simboli diversi

Alternative Considerate array difference



Differenza tra simboli diversi



Differenza tra simboli uguali

Alternative Considerate

array difference

- Fortemente dipendente dall'inclinazione del template e della carta
- Fortemente dipendente dalla binarizzazione
- Fortemente dipendente dalle condizioni di luce
- Anche un singolo pixel diverso fa molte differenza
- Poco preciso rispetto al Template Matching

Alternative Considerate

Momenti di HU

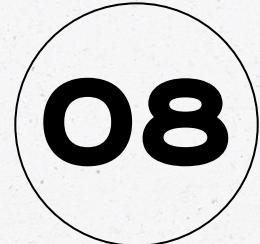
E' stata valutata anche l'opzione di calcolare i **descrittori** dei template e dei simboli tramite i **momenti di HU**.

Per ognuno dei 7 descrittori, si calcolava la differenza con la distanza euclidea.

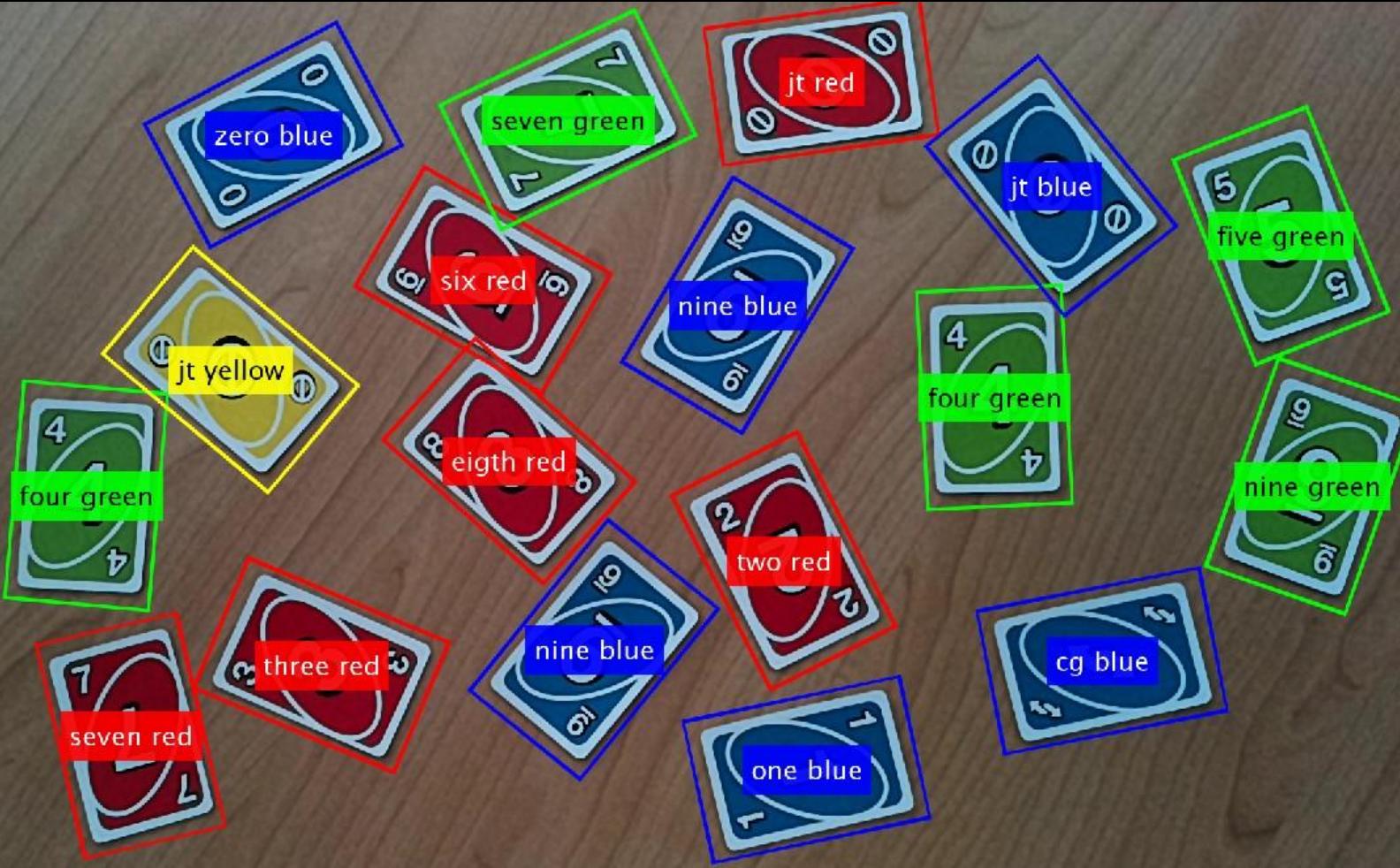
Dato che si basava sulla binarizzazione dei simboli, molto sensibile alle variazioni di luce, abbiamo scartato questa opzione

Ground Truth	background	1757461	111	109	53	13	
red	3085	124838					
green	2212		92778				
yellow	534			30535			
blue	696				30143		
other	31032						

100.0%	0.0%
97.6%	2.4%
97.7%	2.3%
98.3%	1.7%
97.7%	2.3%
	100.0%



ANALISI dei RISULTATI e CONCLUSIONI



RISULTATI OTTENUTI

Immagine 6



Ci sono casi in cui tutte le carte sono riconosciute perfettamente

Immagine 15



Riconosciuti oggetti sconosciuti e alcune carte con un riflesso significativo

Immagine 19



Riconosciuti anche casi di carte che si toccano

COLOR ACCURACY

Training

97,19%

Testing

97,92%

SYMBOL ACCURACY

95,03%

96,24%

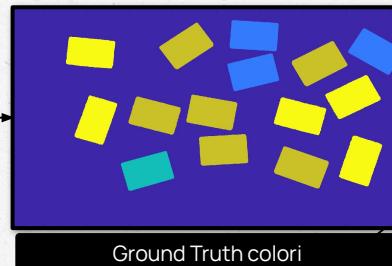
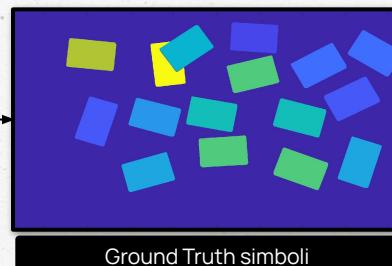
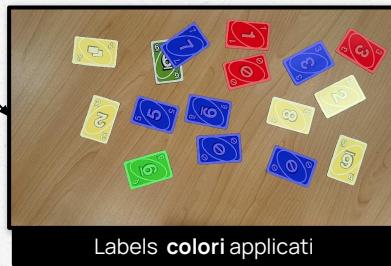


GROUND TRUTHS

Come abbiamo valutato il nostro classificatore?

1. Creando le nostre Ground Truths grazie a **MATLAB Image Labeler**
2. Analizzando l'accuratezza con le **Matrici di Confusione**

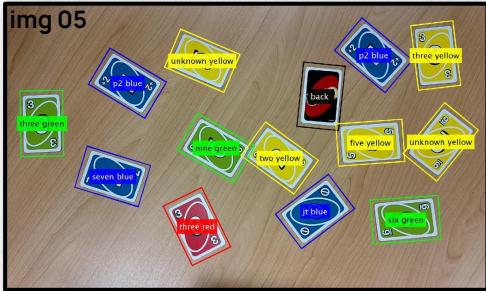
Immagine 20





OTTIMI RISULTATI

Esempi di immagini che hanno avuto ottimi risultati



riconosciuti tutti i colori anche in condizioni di luci non **uniformi**



immagine molto **scura**
(ma riconosciuti lo stesso tutti i colori)



riconosciuti tutti i simboli

Immagine 05

		uno-test-05			
		background	yellow	blue	green
Ground Truth	background	1676054	835	775	509
	yellow	478	149413		
	blue	485		122075	
	green	633			91674
	red	25			30530



		Classificate				
		background	yellow	blue	green	red
Ground Truth	background	99.9%	99.4%	99.4%	99.4%	99.6%
	yellow	0.1%	0.6%	0.6%	0.6%	0.4%

Color classes

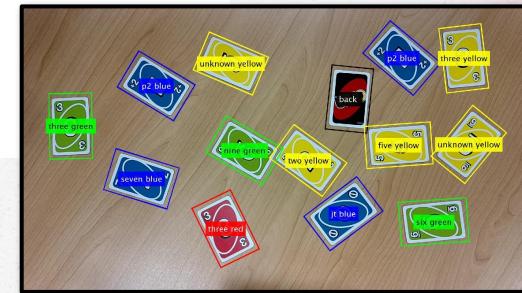
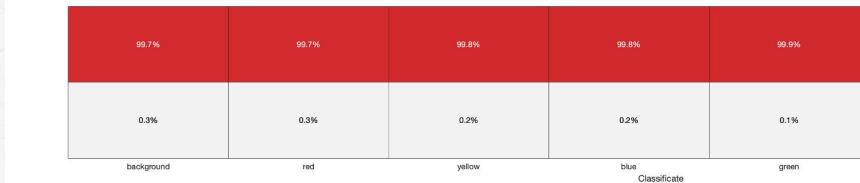


Immagine 25

					uno-test-25	
Ground Truth	background	1702190	445	197	129	68
	red	1736	151026			
	yellow	1248		88286		
	blue	1180			63652	
	green	795				62647

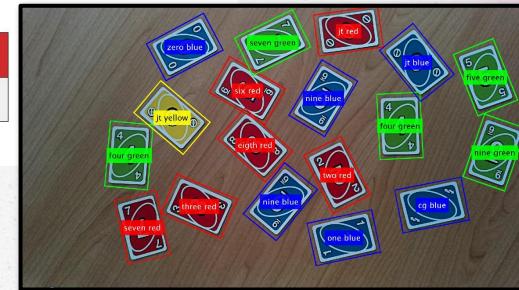


Color classes



Immagine 30

uno-test-30																
Ground Truth	background	1519092	665	500	395	608	161	284	111	318	280	251	362	304	99.7%	0.3%
nine	nine	621	91893											99.3%	0.7%	
jt	jt	673		89925										99.3%	0.7%	
four	four	493			61462									99.2%	0.8%	
seven	seven	322				60505								99.5%	0.5%	
cg	cg	286					32151							99.1%	0.9%	
one	one	183						31561						99.4%	0.6%	
five	five	378							30821					98.8%	1.2%	
three	three	63								30305				99.8%	0.2%	
two	two	57									30109			99.8%	0.2%	
zero	zero	139										29825		99.5%	0.5%	
eighth	eighth	67											29216		99.8%	0.2%
six	six	109												99.6%	0.4%	



Symbol classes

Immagine 12

99.9%	99.1%	99.3%	99.5%	99.0%	98.0%	99.3%	99.4%	99.6%	98.6%	99.0%
0.1%	0.9%	0.7%	0.5%	1.0%	2.0%	0.7%	0.6%	0.2%	3.2%	1.0%
background	cg	j1	p2	egith	zero	one	Classificate	three	two	p4



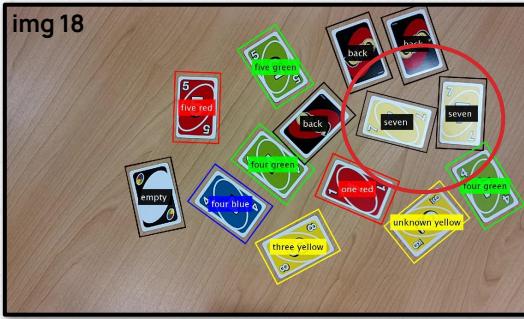
Symbol classes



RISULTATI NON OTTIMALI

Per il colore

img 18



Esempio di immagini che non hanno avuto risultati ottimali per via di un **eccessivo** riflesso su alcune carte.

img 20



Immagine 18

		uno-test-18					
		background	green	yellow	red	blue	other
Ground Truth	background	1727447	2489	2851	2688	970	
	green	73	91452				
	yellow	1		62444			
	red				60110		
	blue	1				30585	
	other	31115		61374			

background	green	yellow	red	blue	other
98.2%	97.4%	49.3%	95.7%	96.9%	

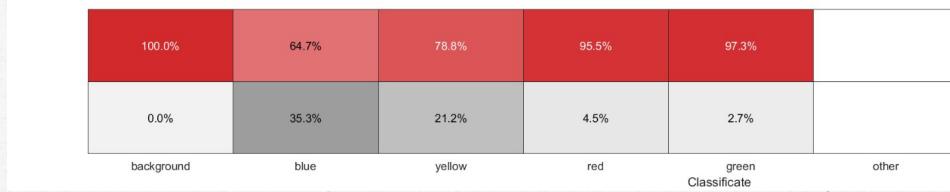
Classificate	background	green	yellow	red	blue	other
	1.8%	2.6%	50.7%	4.3%	3.1%	

Color classes



Immagine 20

		uno-test-20					
		background	blue	yellow	red	green	other
Ground Truth	background	1610723	35424	1849	4185	837	
	blue	5	120594				
	yellow	223		120311			
	red				88013		
	green					30656	
	other	59	30264	30457			



Color classes

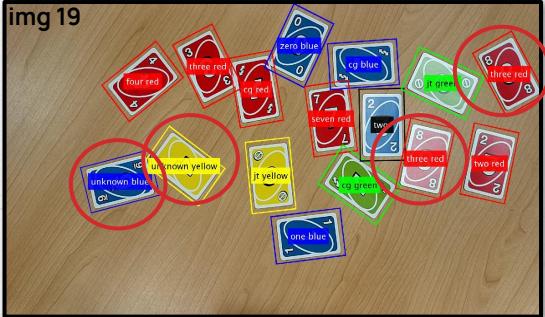




RISULTATI NON OTTIMALI

Per i simboli

img 19



img 15



Esempio di immagini che non hanno avuto risultati ottimali per via della presenza di carte con simboli più **difficili** da riconoscere.

Immagine 19

uno-test-19													
Ground Truth	background	1574583	3144	3314	3091	1635	2495	1348	1767	1458	2777	1724	
cg	186	88328						55					
two			60681										
jt				59617									
one					30549								
four						29225							
zero	11						29125						
seven		4						29097					
three	729	148							28310		60213		
cc						6							
eighth													
gc		60											
nine													
unknown					29424							30496	

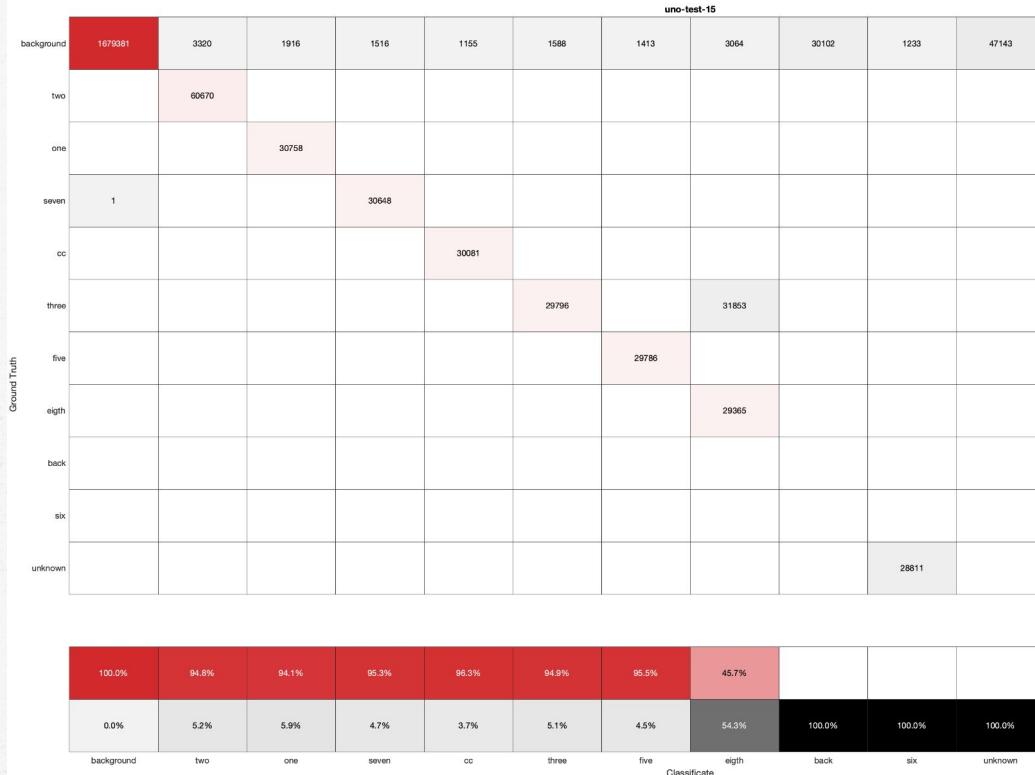
	99.9%	96.3%	94.8%	95.1%	94.9%	47.8%	95.4%	94.3%	95.1%				
background	0.1%	3.7%	5.2%	4.9%	5.1%	52.2%	4.6%	5.7%	4.9%		100.0%		100.0%
cg													
two													
jt													
one													
four													
zero													
seven													
three													
cc													
eighth													
gc													
nine													
unknown													

Classificate

Symbol classes



Immagine 15



Symbol classes

RISULTATI DELLE IMMAGINI PIÙ PROBLEMATICHE



Ricordiamo quali immagini



RISULTATI DELLE IMMAGINI PIÙ PROBLEMATICHE



Ricordiamo quali immagini

img 16



presenza del quaderno giallo

img 24



forte luminosità sulle carte “back”

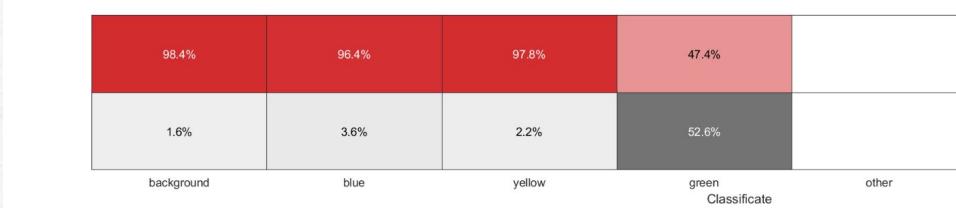
img 17



presenza dei **gruppi di carte**

Immagine 16

		uno-test-16			
		background	blue	yellow	green
Ground Truth	background	1885316	2293	697	33330
	blue		60607		
	yellow	62		30841	
	green				29999
	other	30455			



Color classes



Immagine 16

uno-test-16								
Ground Truth	background	cg	p2	cc	two	nine	three	unknown
background	1883149	637	1739	838	785	1656	32832	
cg	54	30849						
p2	12		30528					
cc	9			30446				
two					30067			
nine								
three								
unknown						29999		

background	cg	p2	cc	two	nine	three	unknown
100.0%	98.0%	94.6%	97.3%	97.5%	100.0%	100.0%	

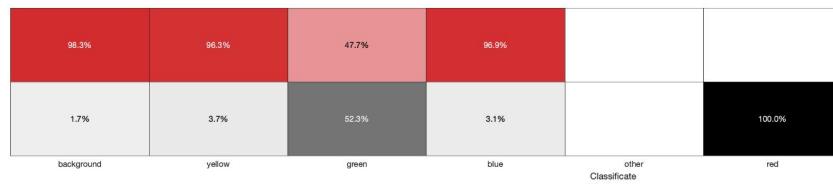
98.0%	2.0%
99.8%	0.2%
100.0%	0.0%
100.0%	0.0%
100.0%	
	100.0%

Symbol classes



Immagine 17

		uno-test-17					
		background	yellow	green	blue	other	red
Ground Truth	background	1819873	3511	33768	974		2082
	yellow	1045	90550				
green			30786				
blue	288			30627			
other	30274					29822	
red							

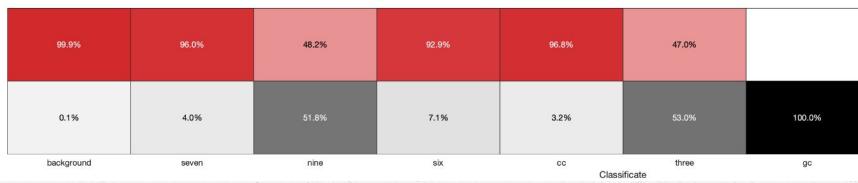
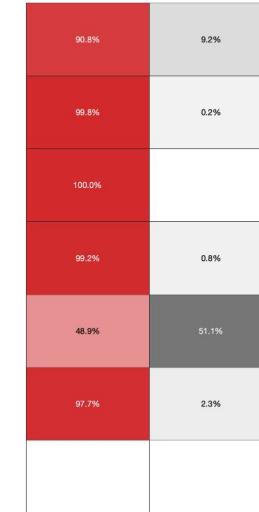


Color classes



Immagine 17

		uno-test-17						
		background	seven	nine	six	cc	three	go
Ground truth	background	1688263	2553	2814	2280	972	33297	130089
	seven	96	61548					13
	nine			30786				
	six	252			29914			
	cc	676		30220		29585		70
	three	574					29531	127
	go							



Symbol classes



Immagine 24

		uno-test-24		
		background	blue	yellow
Ground Truth	background	1983539	139	512
	blue	165	29459	
	yellow			28909
	other	30877		



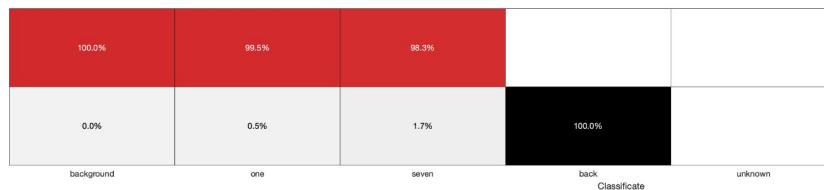
	background	blue	yellow	Classificate
background	98.5%	99.5%	98.3%	
blue	1.5%	0.5%	1.7%	

Color classes



Immagine 24

		uno-test-24			
		background	one	seven	back
Ground Truth	background	1734802	139	512	248737
	one	165	29459		
	seven			28909	
	back				
	unknown	23			30854



Symbol classes



PROBLEMA 1

Il Template Matching non ha una precisione assoluta. È influenzato dalla **nitidezza** delle immagini

SOLUZIONE

Cambiare il **template** dei simboli con scarsa probabilità di riconoscimento

Segmentare i simboli delle carte, estraendo **simboli** con qualità più alta

Aumentare la nitidezza delle carte con un **filtro di sharpening**



Plot carta 9 img 21



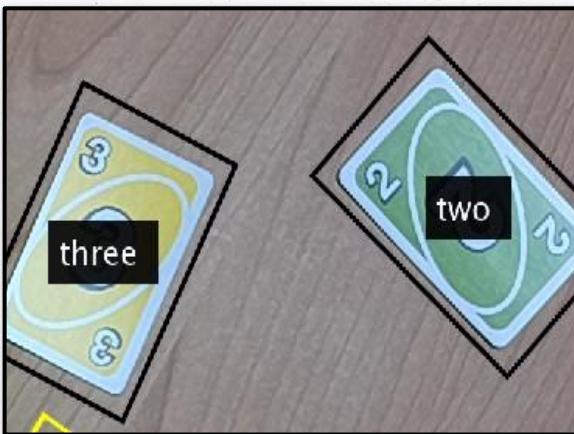
Plot carta 8 img 6



Plot cambio colore img 2

PROBLEMA 2

Le carte sottoposte ad **alta intensità di luce** hanno un colore troppo chiaro che il classificatore difficilmente distingue



SOLUZIONE

Espandere il **dataset** di training di ogni colore in modo da aumentare la **varianza** della tonalità **del colore** presente.

PROBLEMA 3

Le diverse condizioni di luce influiscono molto sulla binarizzazione del canale y per estrarre i bordi dai gruppi di carte.

Per esempio possono generarsi **bordi non definiti**, rendendo l' ovale privo di label propria



Plot *rgb* gruppo di carte 2 Immagine 17



Plot *labels* 17 con l' ovale della carta 3 non distinguibile

SOLUZIONE

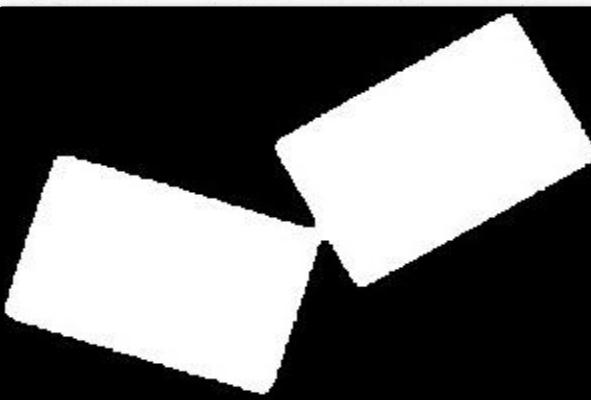
Sistemare la **binarizzazione** e/o usare la **morfologia matematica** per evitare che regioni della carta diverse si tocchino.

PROBLEMA 4

Le carte girate all'interno di un gruppo di carte non vengono estratte.



Plot gruppo di carte girate



Plot maschera gruppo di carte

SOLUZIONE

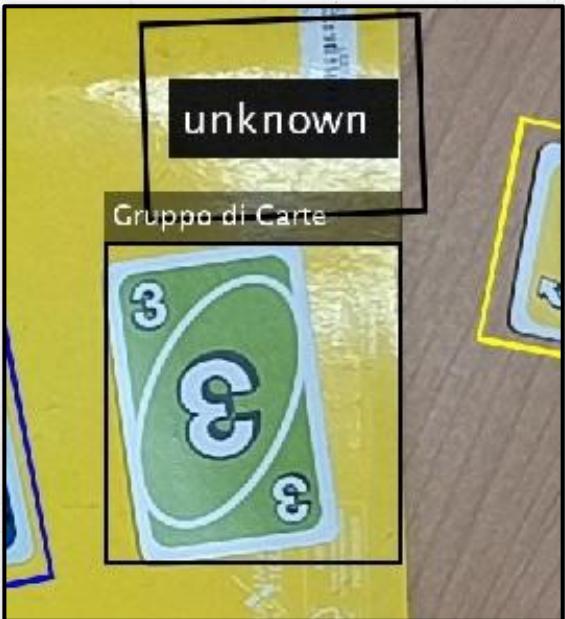
Identificare l' ovale delle carte girate tramite **template matching** con la N per estrarre il **centroide** e applicare la **trigonometria** per estrarre le labels



Plot labels

PROBLEMA 5

Il riflesso di luce viene preso in considerazione come se fosse un oggetto.



Riflesso di luce sul quaderno

SOLUZIONE

Migliorare la lavorazione con i canali dell'immagine in modo che la luce venga presa meno in considerazione.

I **riflessi** principali vengono tolti se l' **area** è **minore** di **AR**

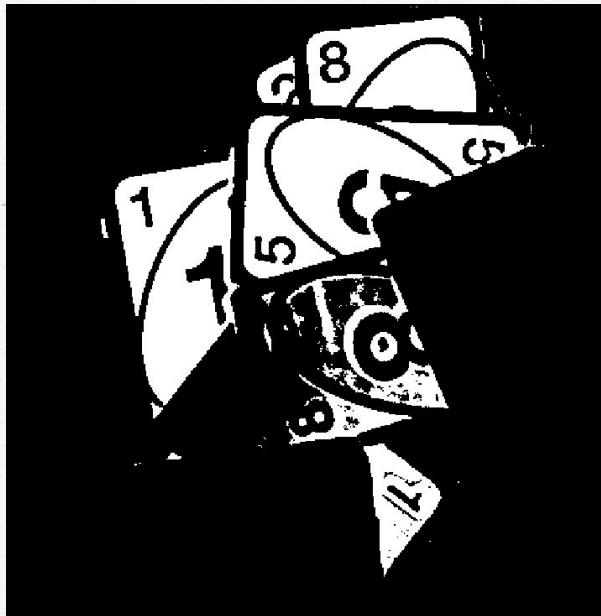
$$AR = \text{area media carta} / 5$$

$$\text{Area media carta} = 3.0081\text{e+04 px}$$

MIGLIORAMENTI PER IL FUTURO



Plot rgb gruppo carte 2 img 17

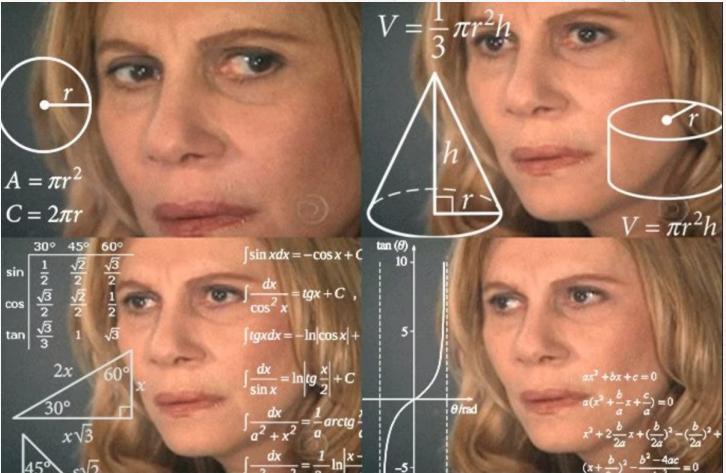


Plot rgb gruppo carte 3 img 17

Identificare il
restante gruppo di
carte tramite ulteriori
**rotate, template
matching e region
growing**

10

APPENDICE

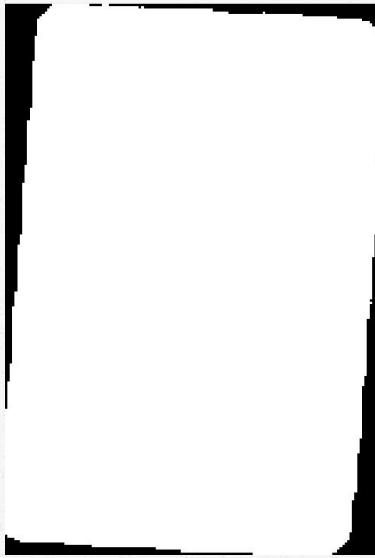




BOUNDING BOX



Plot carta di esempio



Plot maschera della carta



Plot di esempio di una box orientata

L'algoritmo di stampa crea delle **bounding box orientate** per le carte singole, mentre delle **bounding box verticali** per i gruppi di carte

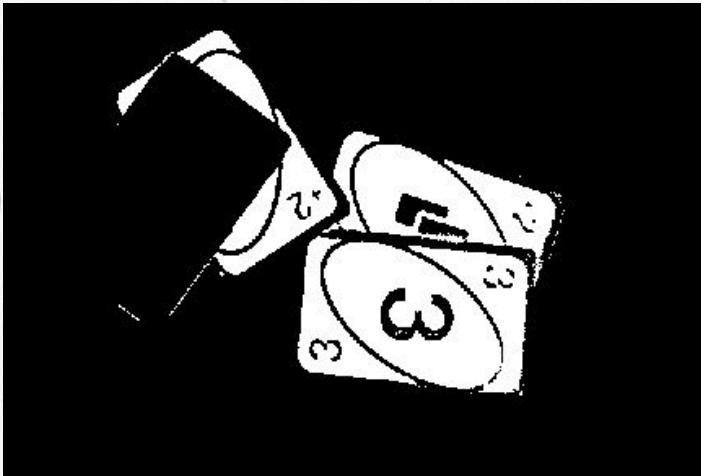
Le bounding box orientate sono così definite: $[c_x, c_y, w, h, \theta] + \text{Symbol} + \text{Color}$

Le bounding box verticali sono così definite: $[x_l, y_m, w, h] + \text{"gc"}$

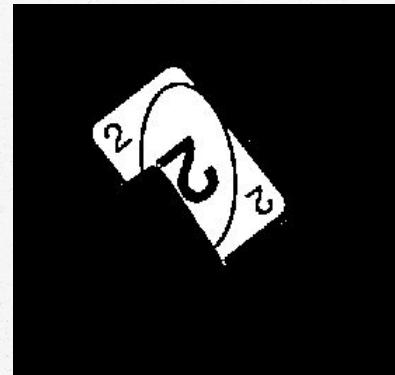
I valori sono calcolati ed estratti dalle rispettive maschere



BOUNDING BOX



Plot gruppo di carte 1



Plot gruppo di carte 2



Plot box gruppi di carte

Dato che non avviene la classificazione sui gruppi di carte, le loro **bbox** saranno di **colore nero**



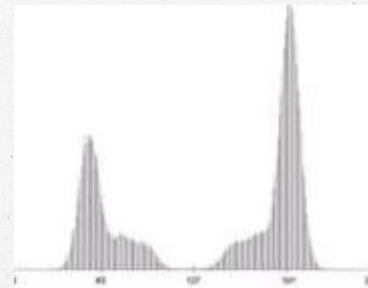
OTSU

Algoritmo che suddivide i pixel in **due gruppi** rispetto ad una soglia che li renda omogenei. L' obiettivo è **minimizzare la varianza intra-gruppo**.

Più la varianza intra-gruppo è piccola e più sono raggruppati **vicino alla media** i due gruppi.

Siccome dal punto di vista computazionale non è ottimale, Otsu ha dimostrato che è più efficiente **massimizzare la varianza inter-gruppo**:

$$\sigma_B^2 = q_1(t) (1 - q_1(t)) (\mu_2(t) - q_2(t))$$



$$q_1(t) = \sum_{i=1} ip(i)$$

$$q_2(t) = \sum_{i=t+1} ip(i)$$

$$\mu_2 = \sum ip(i)\mu/q_1(t)$$



PREWITT

E' un filtro derivativo di primo ordine utilizzato per individuare gli edge all' interno di un immagine

E' composto da 2 kernel, uno per gli edge verticali ed uno per quelli orizzontali, così strutturati:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \otimes [-1 \ 0 \ +1] = \begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix}$$

Kernel edge verticali

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} -1 \\ 0 \\ +1 \end{bmatrix} = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Kernel edge orizzontali

Sono generati dalla convoluzione associativa del filtro derivativo con il filtri di media.

Il filtro di Prewitt, dunque, applica anche un leggero smoothing.

E' importante sottolineare che inspessisce l' edge, rendendolo doppio (2px)



SOBEL

E' un filtro derivativo di primo ordine utilizzato per individuare gli edge all' interno di un immagine

E' composto da 2 kernel, uno per gli edge verticali ed uno per quelli orizzontali, così strutturati:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \otimes [-1 \ 0 \ +1] = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$$

Kernel edge verticali

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} -1 \\ 0 \\ +1 \end{bmatrix} = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Kernel edge orizzontali

Sono generati dalla convoluzione associativa del filtro derivativo con il filtro di smoothing gaussiano

Il filtro di Sobel, dunque, applica anche uno smoothing dando maggior peso al pixel centrale.

E' importante sottolineare che inspessisce l' edge, rendendolo doppio (2px)



ROBERTS

E' un filtro derivativo di primo ordine utilizzato per individuare gli edge all' interno di un immagine

E' composto da 2 kernel, creati per calcolare il gradiente obliquo

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Kernel edge 1

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Kernel edge 2

Il difetto è che, essendo 2×2 , l' applicazione di esso non avviene nel centro del filtro.

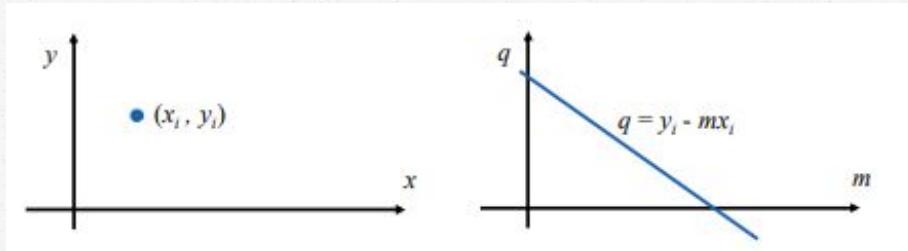


HOUGH

Dato un punto nel piano cartesiano X e Y, esistono infinite rette passanti per esso: tutte quelle che soddisfano l'equazione $y = mx + q$.

Girando l'equazione nella forma $q = y - mx$, si ottengono tutte le coppie di valori (m, q) che le rette possono assumere affinché passino per il punto X e Y.

L'insieme di questi punti rappresenta una retta a sua volta nello **spazio dei parametri**



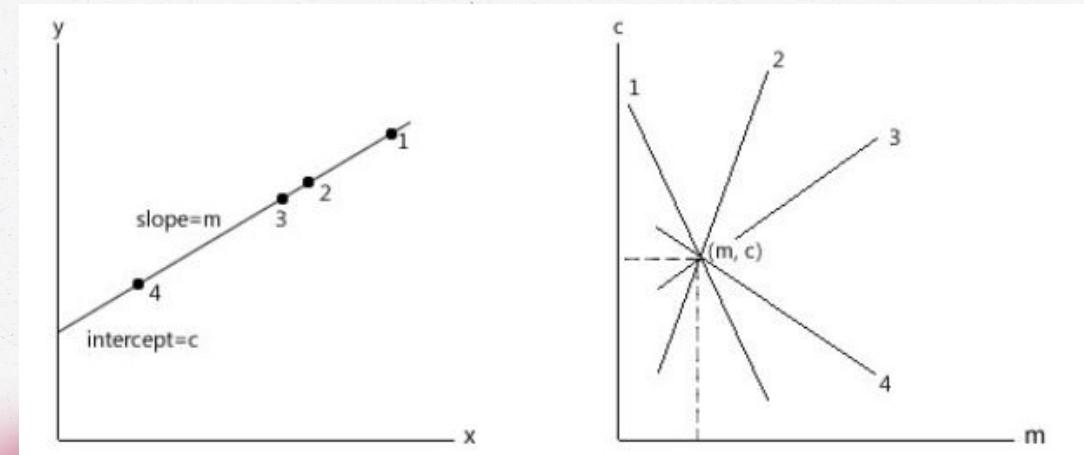


HOUGH

N punti nello spazio allineati generano **n rette dei parametri** che si intersecano in un punto (m_o, q_0) .

Questo punto indica la coppia di valori (m, q) che una generica retta nella forma $y = mx + q$ deve avere per poter passare in tutti gli n punti allineati.

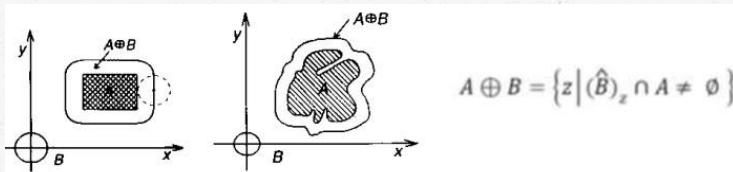
Per individuare le rette nella maschera della carta, dunque, si generano tutte le rette dei parametri per ogni punto dell'immagine e si considerano i punti di maggior intersezione tra esse, **picchi**.





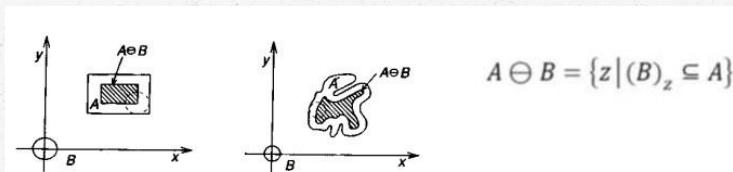
MORFOLOGIA

Dilation: L'insieme dei punti di riferimento p, per cui B_p ed A hanno almeno un punto in comune.



$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\}$$

Erosion: L'insieme dei punti di riferimento p, per cui B_p è interamente contenuto in A.



$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

Opening: L'apertura di A con B è la successione di una *erosion* con B ed di una *dilation* con B.

$$A \circ B = (A \ominus B) \oplus B$$



MORFOLOGIA

Closing: La chiusura di A con B è la successione di una dall'espansione (dilation) con B, seguita dalla erosione (erosion) con B

$$A \bullet B = (A \oplus B) \ominus B$$

Filling by Dilation: Se c'è presenza di una o più regioni di sfondo circondate da una connessione di elementi di foreground.

Sia A un insieme contenente 8-connected borders che racchiude la regione dei buchi, che devono essere riempiti, la sequenza X_0, \dots, X_k può essere costruita, dove X_0 è un insieme contenente un punto di ogni foro e X_j è definito come:

$$X_j = (X_{j-1} \oplus B) \cap A^c$$

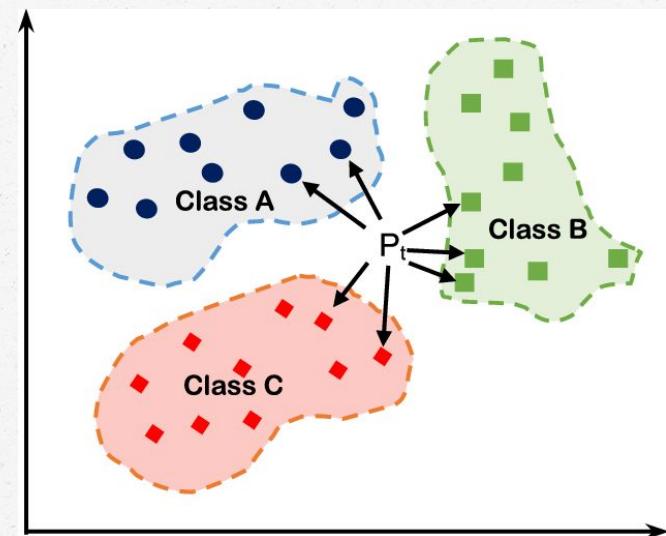


CLASSIFICATORE KNN

Algoritmo di Machine Learning supervisionato utilizzato in questo contesto come classificatore.

Dopo l'addestramento del classificatore, per assegnare una classe a un punto, si individuano i k punti più vicini e si attribuisce al punto la classe più frequente tra questi.

Dati due punti, la distanza tra questi può essere calcolata secondo diversi algoritmi: distanza euclidea, distanza cosine, ecc...



MOMENTI DI HU

Si definiscono momenti mn-iesimi di un'immagine, le grandezze che descrivono l'immagine. In particolare, se è presente un solo oggetto, esso è identificato proprio da esse.

$$m_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} x^p y^q g(x, y)$$

$$\begin{aligned} M_{00} &= \text{Area} \\ M_{01} &= i\text{-baricentro} \\ M_{01} &= j\text{-baricentro} \end{aligned}$$

I momenti centrali, invece, sono calcolati considerando il **baricentro**. Ciò li rende **invarianti** rispetto alla traslazione

$$M'_{mn} = \sum_{i=0}^{n_r-1} \sum_{j=0}^{n_c-1} (i - i_b)^m (j - j_c)^n F(i, j)$$

Infine, i momenti di Hu sono 7 **descrittori invarianti per rotazione, scala e traslazione**, ottenuti **normalizzando** i momenti centrali

$$V_{mn} = \frac{M'_{mn}}{(M_{00})^\alpha}, \quad \alpha = \frac{m+n}{2} + 1$$

Normalizzazione dei momenti centrali

ALGORITMO DI BRESENHAM

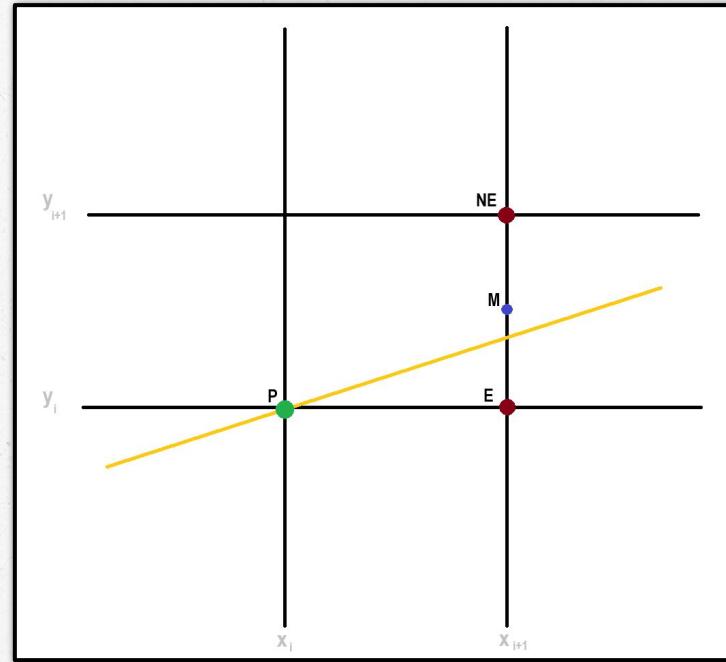
In base alla posizione di **M** rispetto alla retta, viene preso il pixel **E** o **NE**.

P: punto di partenza = (x_i, y_i)

E: pt. possibile successivo = (x_{i+1}, y_i)

NE: pt. possibile successivo = (x_{i+1}, y_{i+1})

M: punto medio = $(x_{i+1}, y_{i+\frac{1}{2}})$



Sauvola

Algoritmo locale, analizza delle regioni locali dell'immagine sulle quali determina la soglia che meglio separa le regioni chiare da quelle scure.

Algoritmo che si adatta al meglio in presenza di ombre e variazioni di luminosità nell'immagine.

- Per ogni pixel nell'immagine, viene calcolata la media dei livelli di grigio in una finestra rettangolare intorno a quel pixel ($m(x,y)$).
- Calcolo deviazione standard dei livelli di grigio nella stessa finestra intorno al pixel ($s(x,y)$).
- Calcolo soglia locale, con k come parametro di regolazione e R valore massimo della deviazione standard:

$$t(x, y) = m(x, y) \left[1 + k \left(\frac{s(x, y)}{R} - 1 \right) \right]$$

- Binarizzazione immagine