Chessboard recognition

Riconoscimento degli schemi di gioco degli scacchi

Elaborazione delle immagini

2017 - 2018

Struttura dell’ applicazione

Il linguaggio di programmazione utilizzato è MatLab.

Funzione principale: ***read_chessboard(****image****)***

Input: immagine RGB (*image*)

La funzione restituisce la stringa FEN rappresentante l’immagine passata in input all’utente tramite la console.

La funzione agisce in 3 passi:

1. Individuazione della scacchiera
2. Correzione prospettica
3. Riconoscimento

L’intero progetto è suddiviso in varie funzioni specifiche per vari aspetti dell’elaborazione in modo da renderlo maggiormente comprensibile e facilmente modificabile.

Premesse

Le immagini, per essere riconosciute correttamente, devono rispondere alle seguenti caratteristiche:

1. La scacchiera deve essere correttamente a fuoco
2. La scacchiera non deve essere nascosta da un qualsiasi oggetto esterno
3. Il foglio contenente la scacchiera non deve essere soggetto ad una rotazione minore di -45° o maggiore di 45° rispetto all'asse delle ascisse
4. Il foglio deve rappresentare buona parte dell'immagine totale
5. Eventuali ombre proiettate sulla scacchiera non devono essere troppo nette
6. Il foglio in prospettiva deve mantenere caratteristiche geometriche tali da non renderlo associabile ad un quadrato

Passo 1: Individuazione della scacchiera

Viene effettuato il bilanciamento del bianco utilizzando il metodo «white spot»

Immagine in input

White Balance

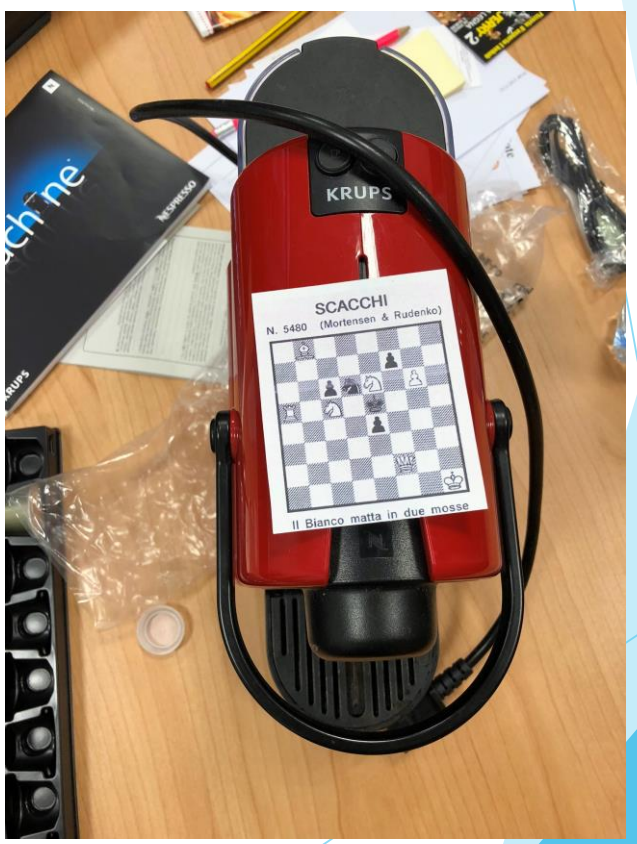


Immagine dopo WB

Passo 1: Individuazione della scacchiera

L’immagine viene passata nello spazio HSV e vengo presi i canali Saturation e Value.

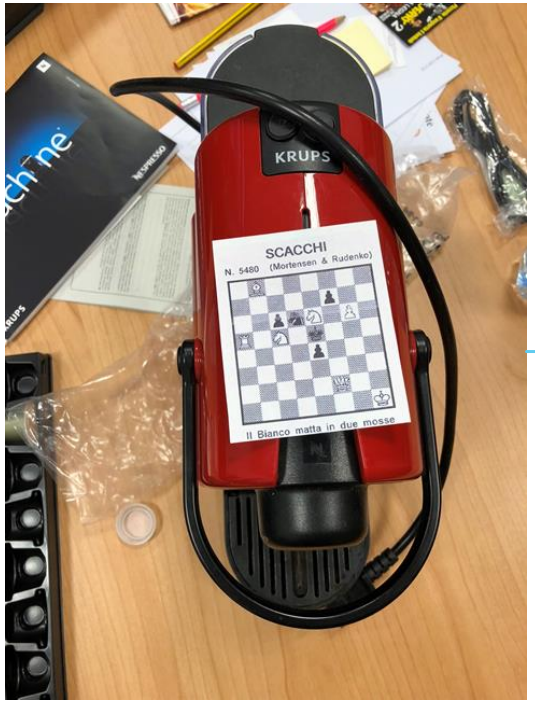
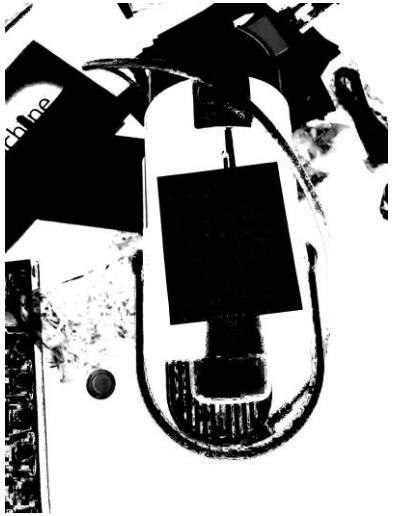


Immagine dopo WB

Saturation channel

Value channel

Passo 1: Individuazione della scacchiera



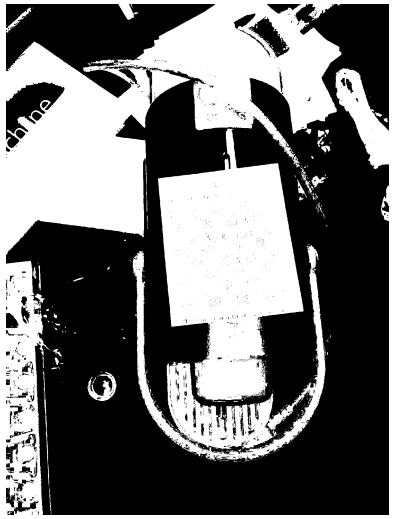
Saturation

Value

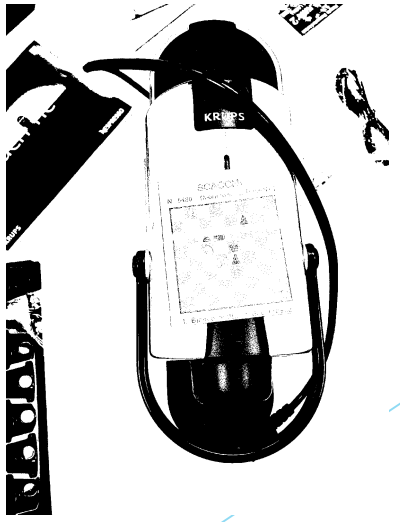
Vengono modificati i valori di intensità delle due immagini in modo da rendere più uniforme la regione della scacchiera

Passo 1: Individuazione della scacchiera

Saturation < 0.15



Value > 0.91



Entrambi i canali vengono sogliati in modo da binarizzarli e rendere la regione d'interesse meglio identificabile

Passo 1: Individuazione della scacchiera

Immagine dopo WB

rgb2gray

Immagine a livelli di grigio

stretching

Stretching dell'istogramma

Passo 1: Individuazione della scacchiera

I bianchi ed i neri vengono compressi e l'immagine viene binarizzata utilizzando l'algoritmo di Sauvola con dimensione della finestra pari a [500 500]

imadjust

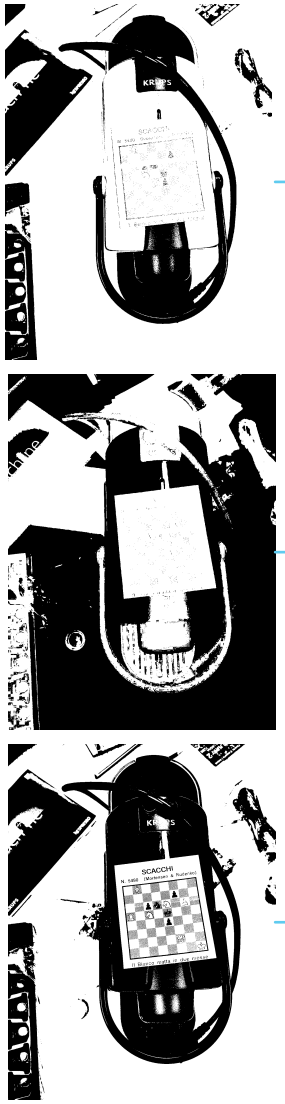
binarizzazione

Passo 1: Individuazione della scacchiera

Value

Saturation

Binarizzata



Le 3 immagini
- Canale value
- Canale saturazione
- Immagine binarizzata

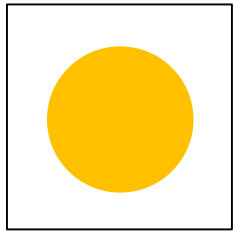
Vengono messi in AND generando così una maschera più precisa della scacchiera.

Attraverso delle prove ho osservato che i canali in cui il foglio appariva sempre identificabile (in tutte le immagini) erano il canale della saturazione e quello del value.

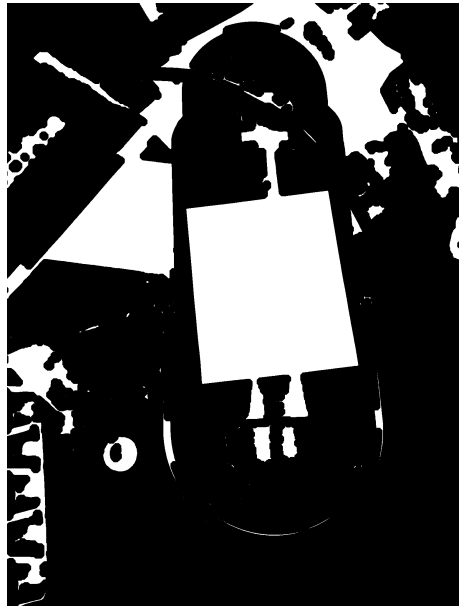
Passo 1: Individuazione della scacchiera

Viene effettuato un closing in modo da rendere l'area del foglio omogenea, e successivamente una opening per la rimozione di aree non di interesse

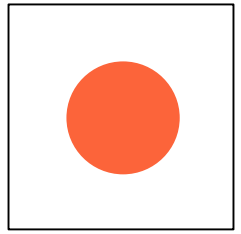
Elemento strutturante



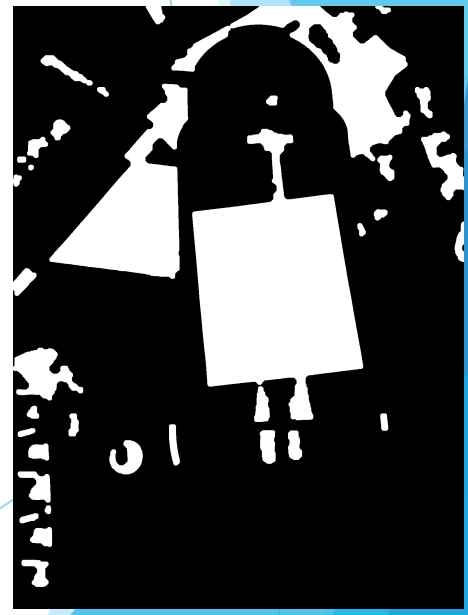
Closing



Elemento strutturante



Opening



La maschera così ottenuta verrà utilizzata tra poco da un'altra funzione per l'individuazione precisa della scacchiera.

Passo 1: Individuazione della scacchiera

Cercheremo ora di individuare precisamente la regione rappresentante la scacchiera.

Trasformo l'immagine originale nella sua versione grayscale, effettuo uno stretching dell'istogramma ed un contrast enhancement per rendere migliore la successiva binarizzazione con sauvola.

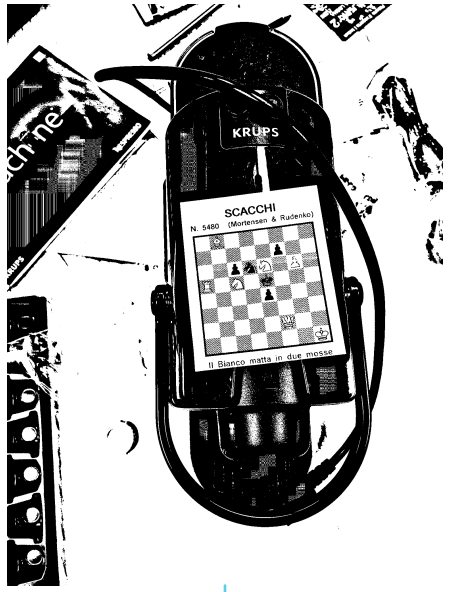
Immagine a livelli di grigio

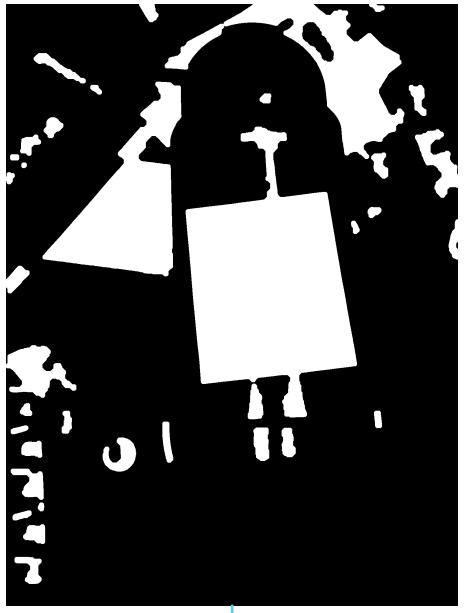
Stretching dell'istogramma

Aumento del contrasto

Binarizzazione Sauvola

Passo 1: Individuazione della scacchiera





X

La maschera trovata precedentemente e l'immagine appena binarizzata vengono combinate in modo ottenere un'immagine binaria che includa la scacchiera e qualche regione spuria rimasta dai passaggi precedenti

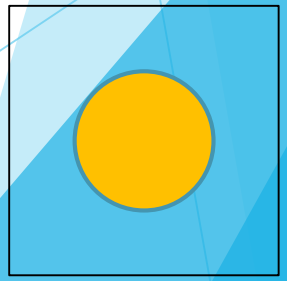
Passo 1: Individuazione della scacchiera

opening

closing

Vengono eseguite in ordine un' opening ed un closing sull'immagine, non in tutte le immagini questo passaggio è necessario, infatti su questa immagine di esempio il raggio dell'elemento strutturante usato **è pari a 0.** queste due operazioni si trovano in un ciclo che incrementa il raggio dell'elemento strutturante finché non si ottiene un risultato valido (una maschera non vuota)

Elemento strutturante



Passo 1: Individuazione della scacchiera

Smoothing

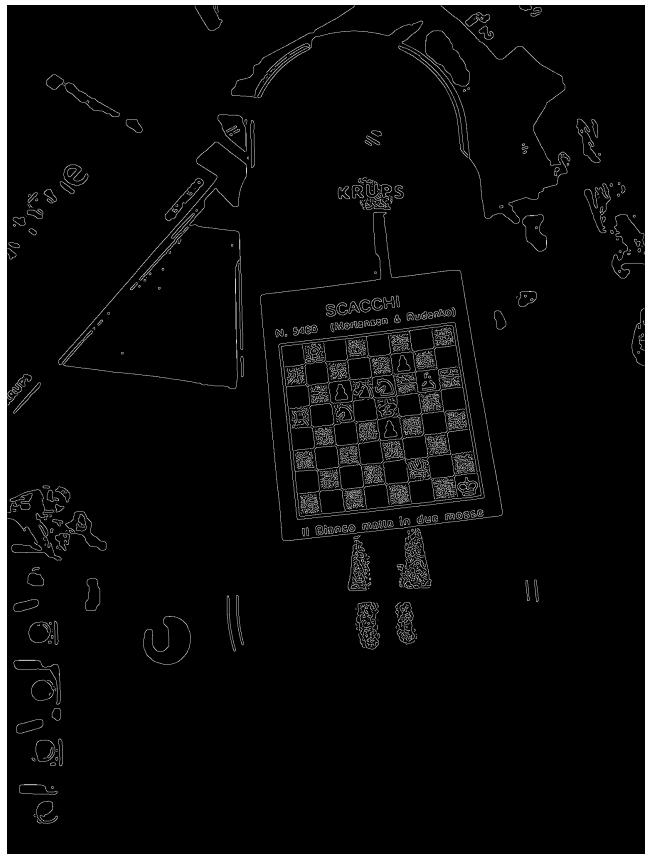
Viene applicato un filtro di smoothing gaussiano con kernel 2x2 in modo da rendere le linee meno seghettate e più facilmente riconoscibili come continue per l'edge detection

Passo 1: Individuazione della scacchiera

Immagine smoothing

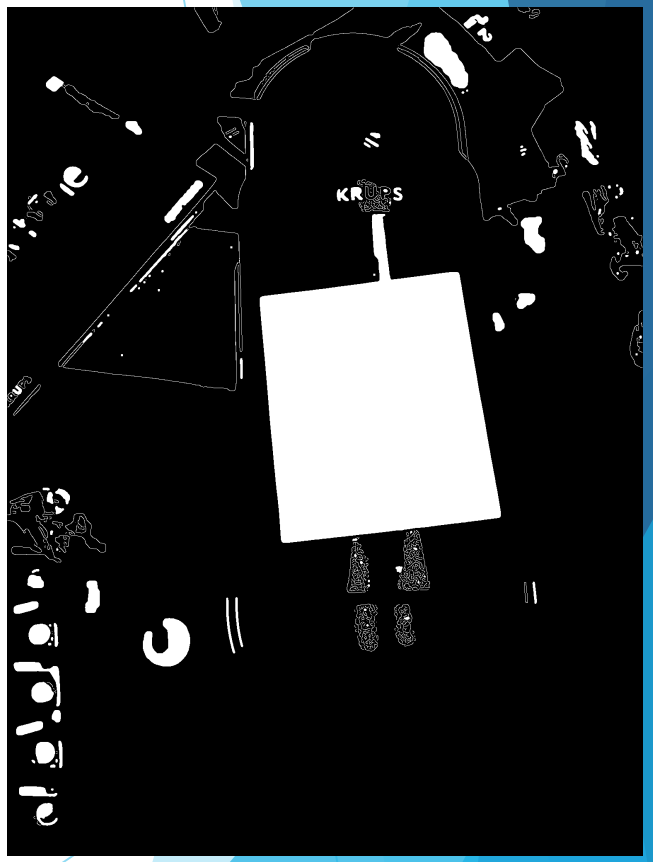
Edge

Immagine degli edge



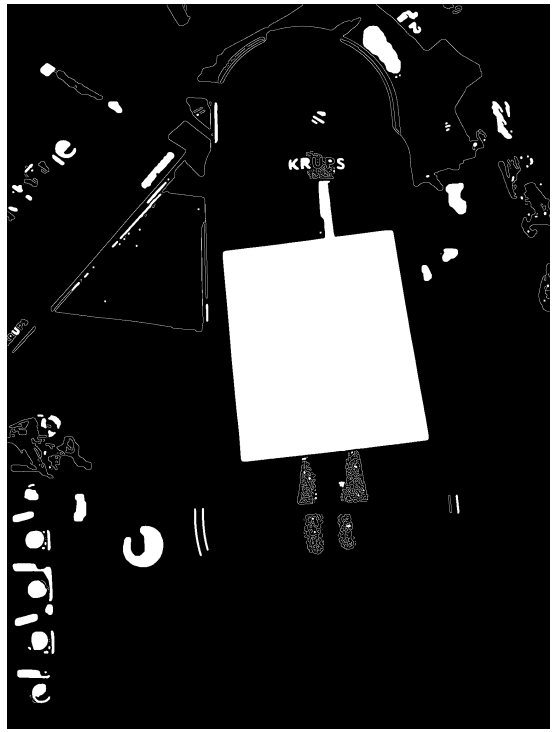
imfill

Risultato imfill

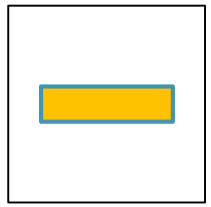


Viene effettuato un edge detection sull'immagine tramite il metodo di Canny, le regioni delimitate (holes) contenute nell'immagine degli edge vengono riempite utilizzando la funzione imfill()

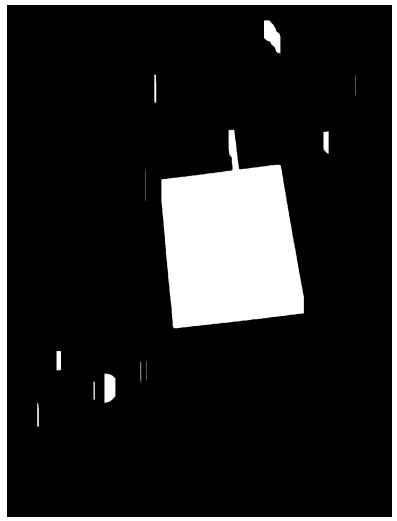
Passo 1: Individuazione della scacchiera



Elemento strutturante

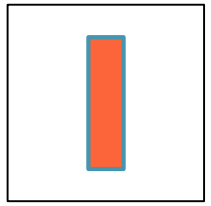


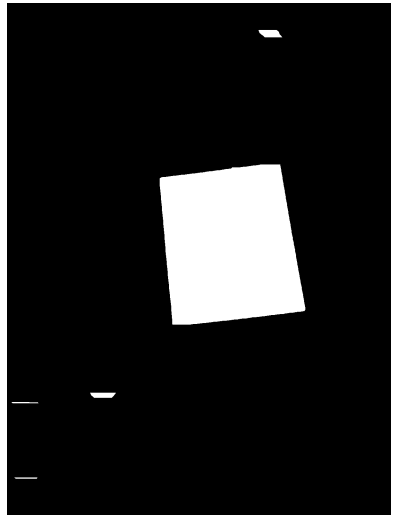
opening



opening

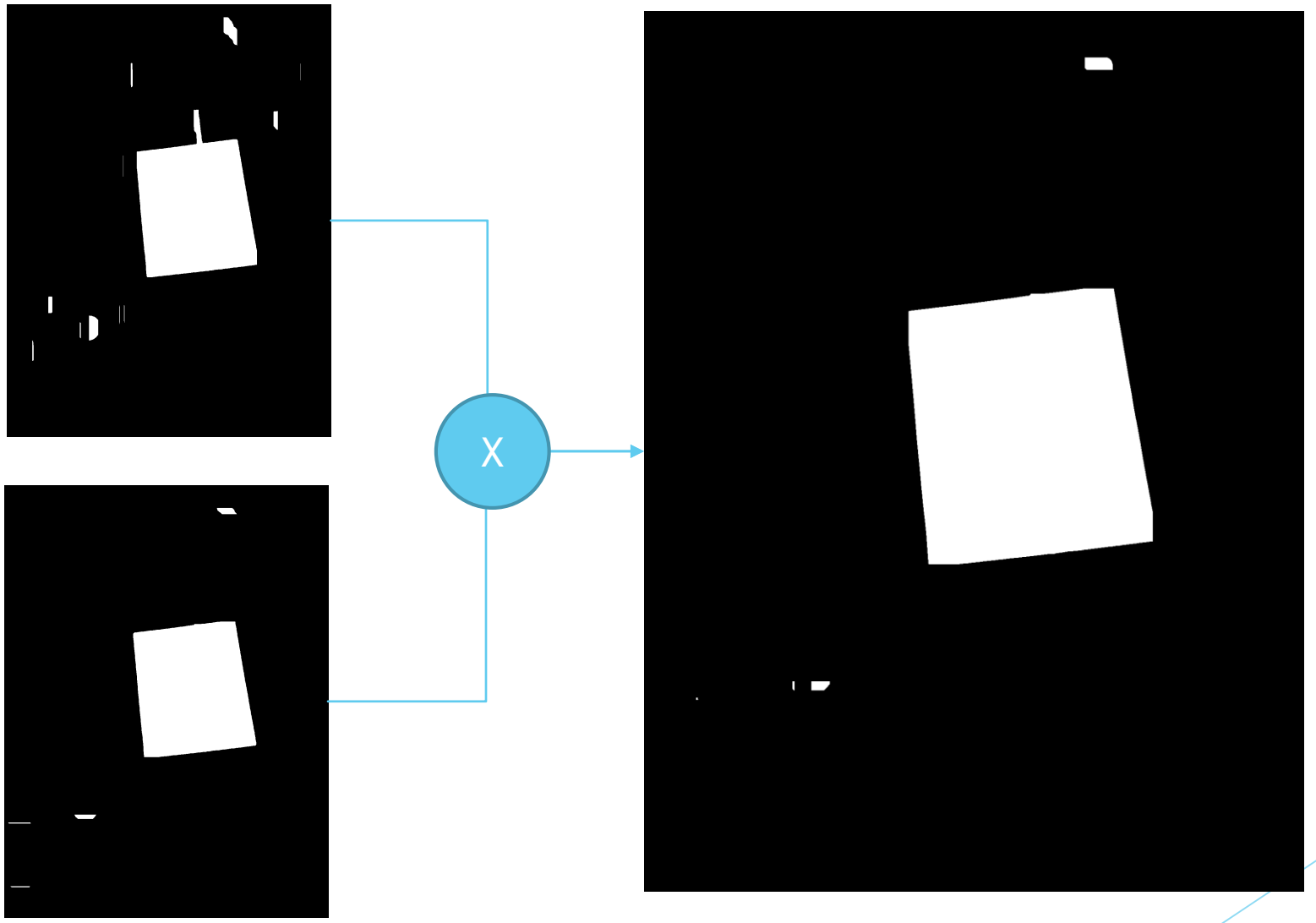
Elemento strutturante





Vengono create due immagini attraverso l'opening della maschera con 2 elementi strutturanti diversi

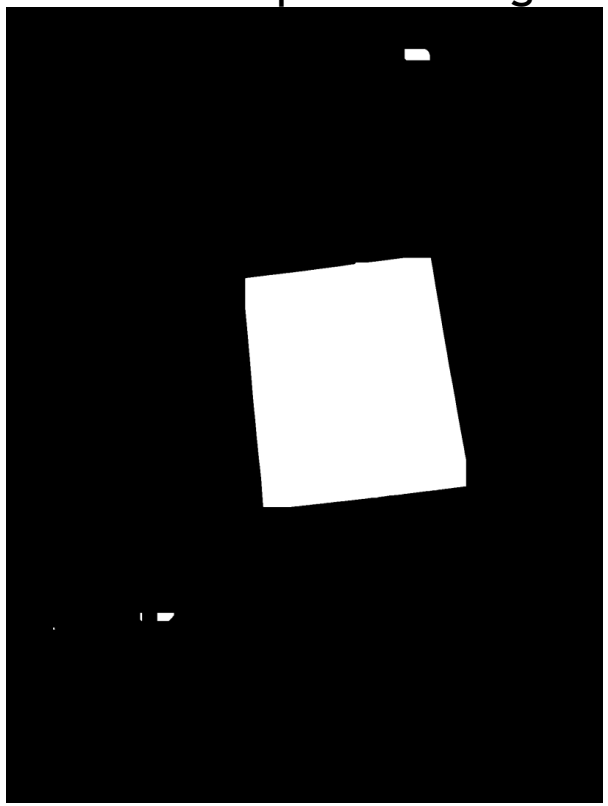
Passo 1: Individuazione della scacchiera



Combinando le due maschere otteniamo una maschera molto più pulita, di contro possiamo notare che gli angoli del foglio sono stati tagliati, correggeremo questo problema più avanti

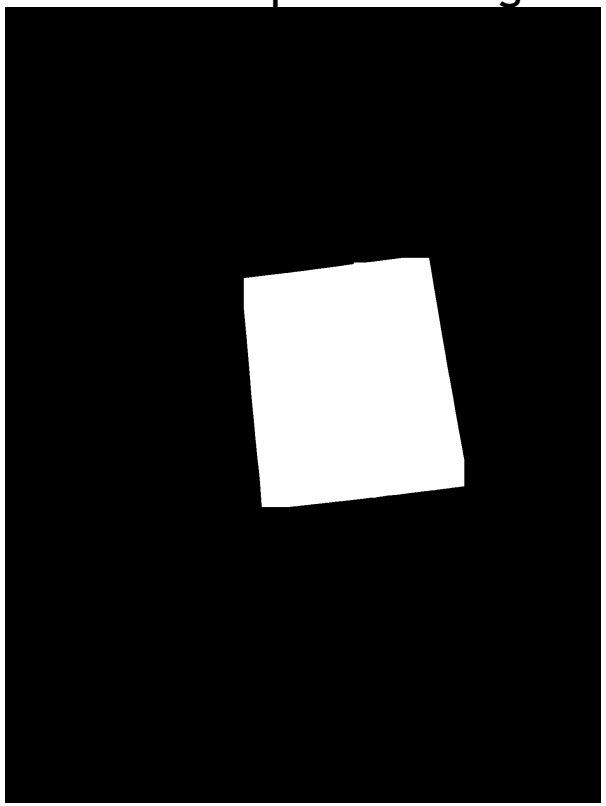
Passo 1: Individuazione della scacchiera

Maschera dopo morfologia

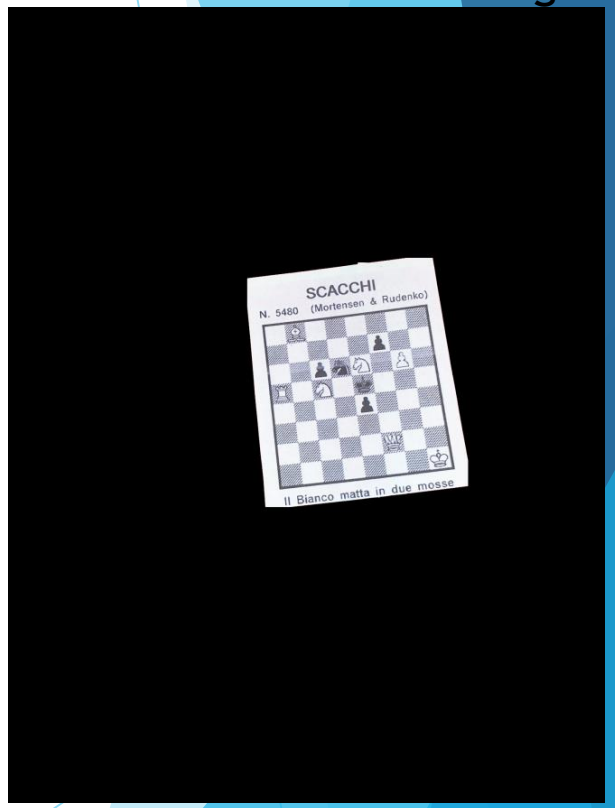


Max label

Maschera dopo labelling

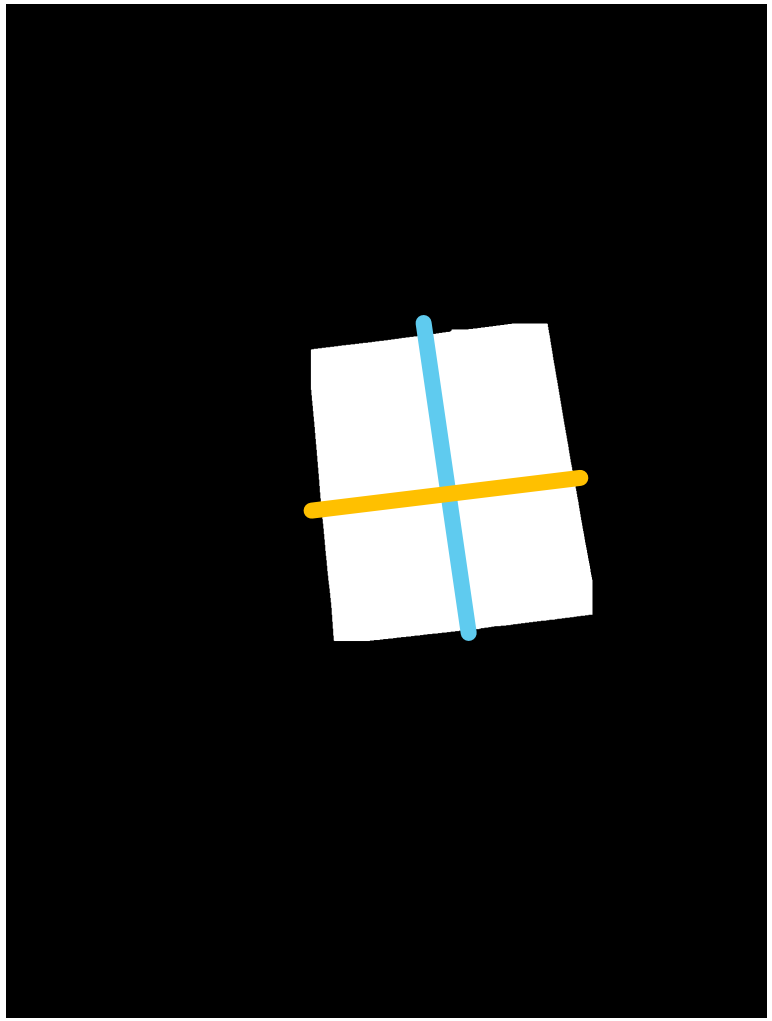


Combinazione con immagine



Viene effettuato il labelling delle CC e viene estratta la CC con area maggiore (scacchiera), dopo di che in base al parametro passato alla funzione viene restituita la maschera (parametro == 0)oppure la combinazione della maschera con l'immagine a livelli di grigio (parametro == 1)

Passo 2: correzioni prospettiche



Asse minore

Asse maggiore

Tramite il rapporto tra Asse minore ed Asse maggiore cerco di capire se la regione trovata rappresenta la sola scacchiera oppure anche il foglio che la contiene (Quadrato vs Rettangolo)

Nel secondo caso, che è il caso di questa immagine, ci ricondurremo ad ottenere la sola scacchiera.

Passo 2: correzioni prospettiche

Tramite le coordinate della BoundingBox croppo l'immagine e ne trovo gli edge con Canny

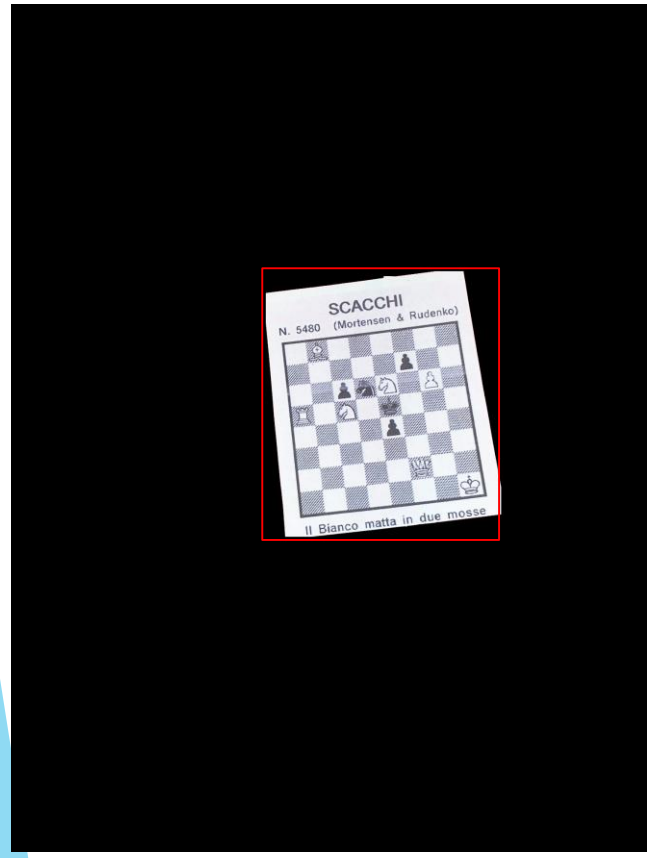


Immagine + BoundingBox

imcrop

Immagine a seguito del crop

edge

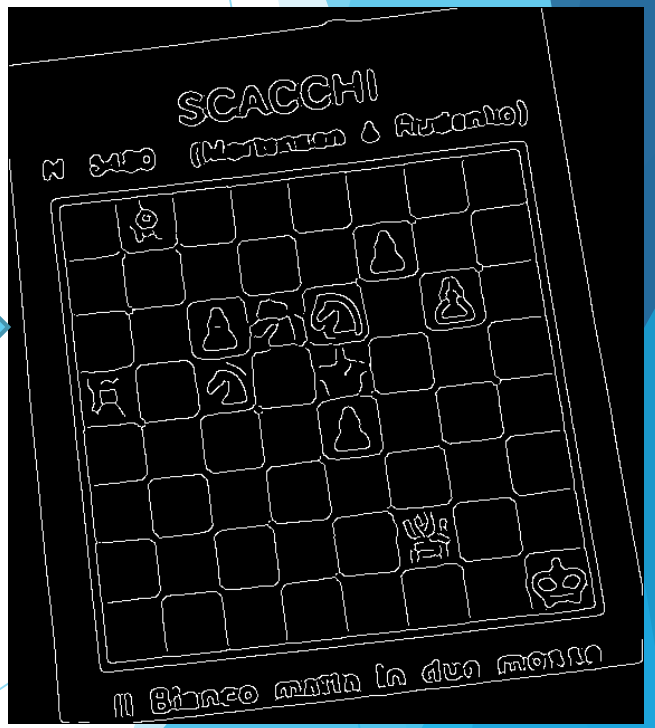


Immagine degli edge

Passo 2: correzioni prospettiche

Le regioni delimitate (holes) contenute nell'immagine degli edge vengono riempite utilizzando la funzione imfill, per eliminare alcune zone non necessarie eseguo una opening. Nel caso l'opening non sia sufficiente viene eseguito un labelling delle componenti connesse e viene presa la cc con area massima.

Elemento strutturante

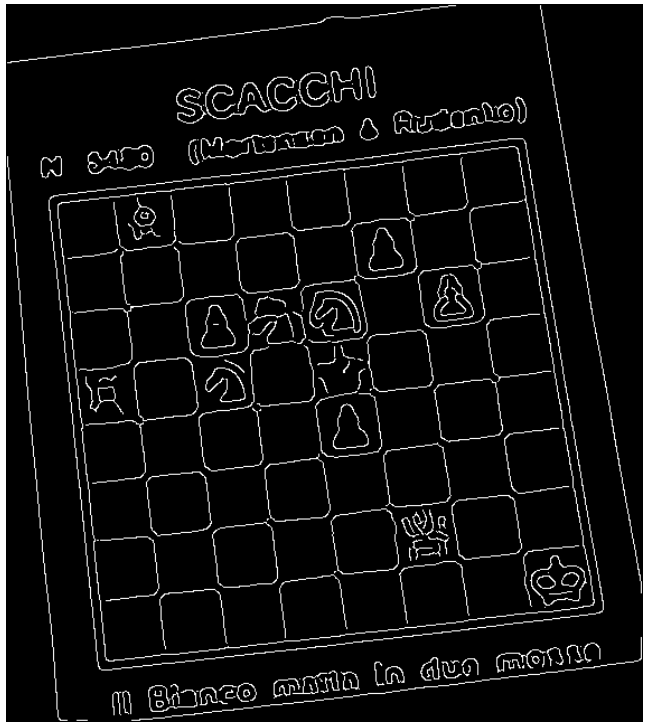
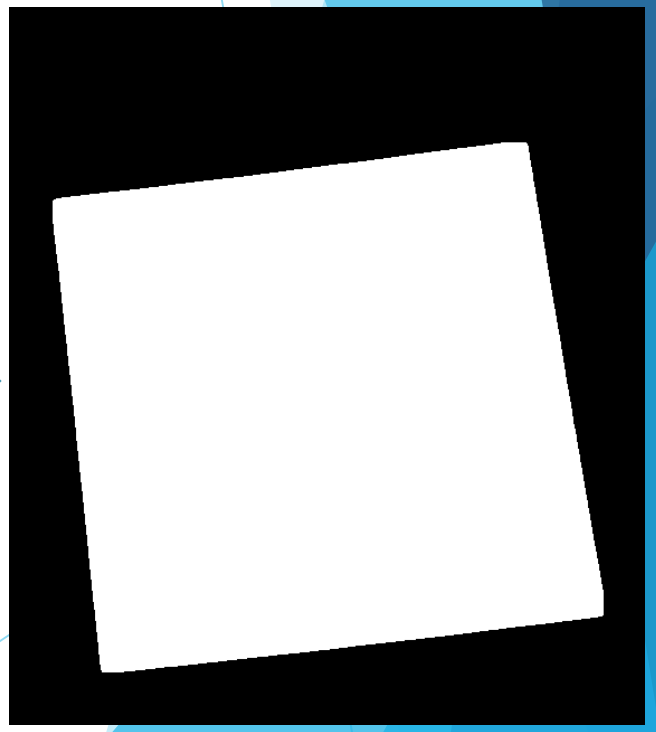


Immagine degli edge

imfill

Immagine degli edge fillata

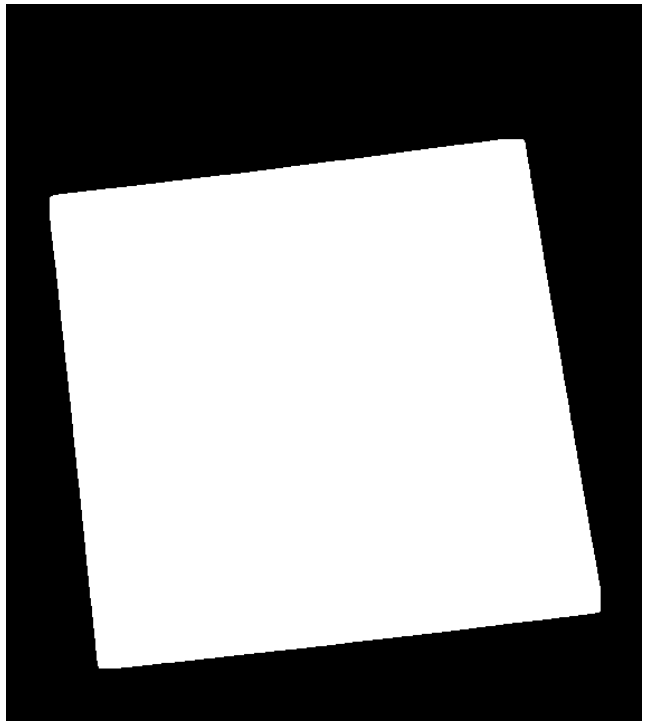
opening



Maschera ripulita

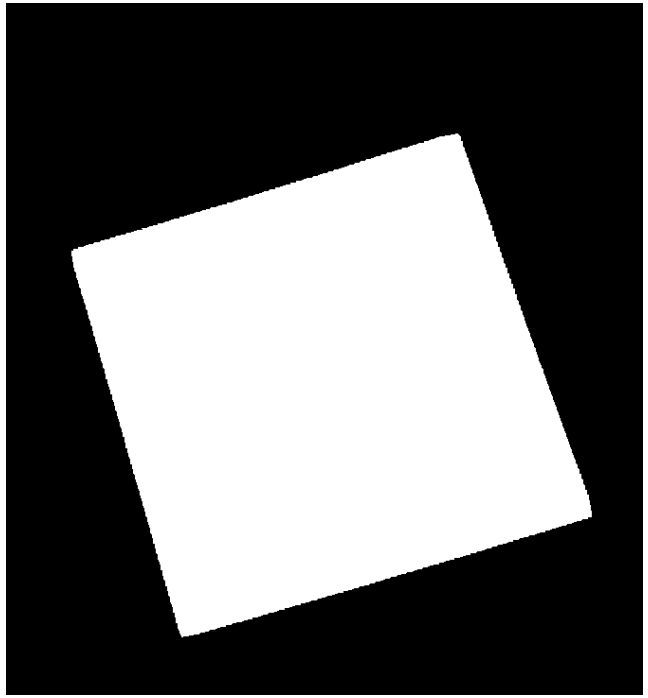
Passo 2: correzioni prospettiche

La maschera viene rotata di 10° in senso antiorario più volte finché la Trasformata di Hough non è in grado di identificare 4 linee



Maschera scacchiera

rotate



Maschera scacchiera ruotata di 10°

edge

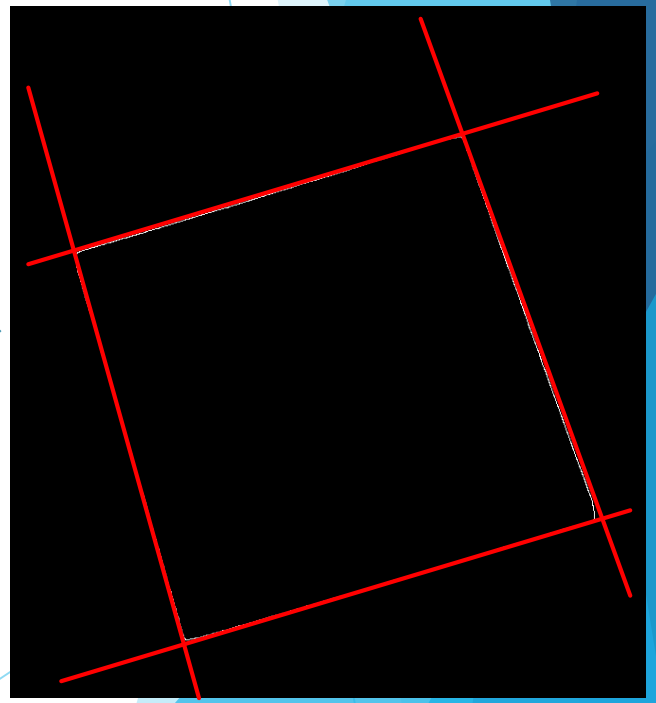
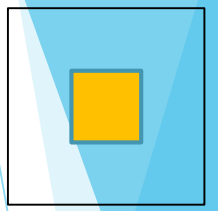


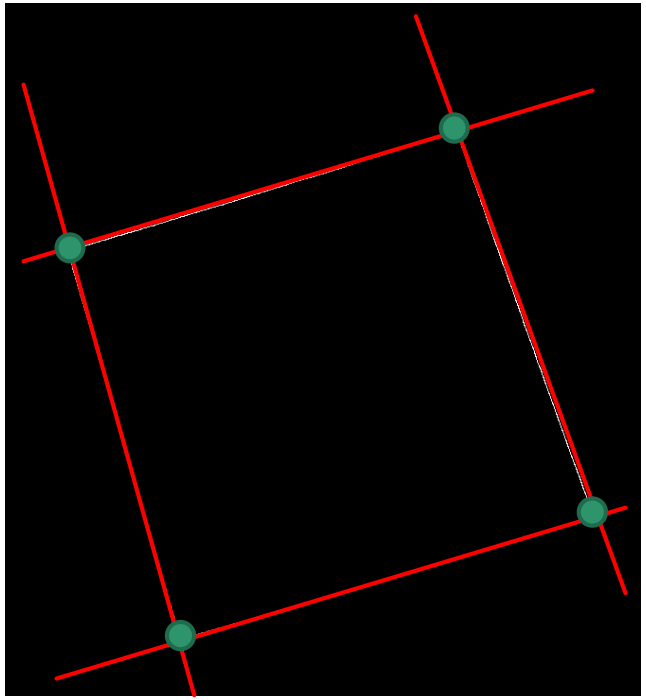
Immagine degli edge + Hough

Passo 2: correzioni prospettiche

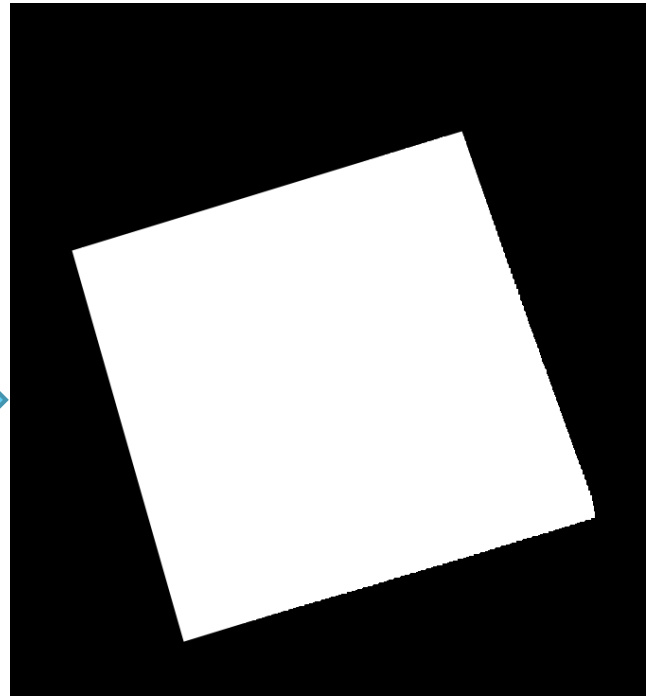
Elemento strutturante



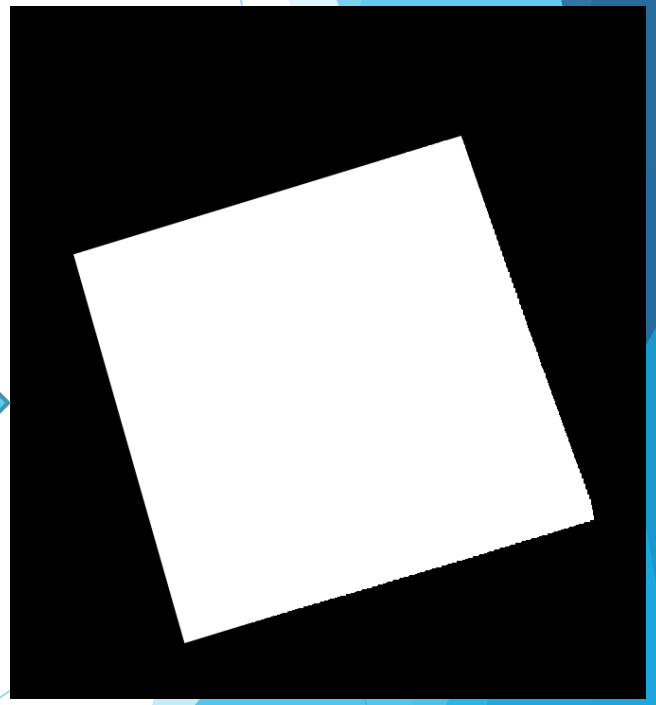
Trovo i punti d'intersezione generati dalle rette trovate con Hough e li uso per inserire una figura tramite insertshape() che rappresenti meglio la maschera della scacchiera, la erodo per renderla più precisa.



shape



erosion

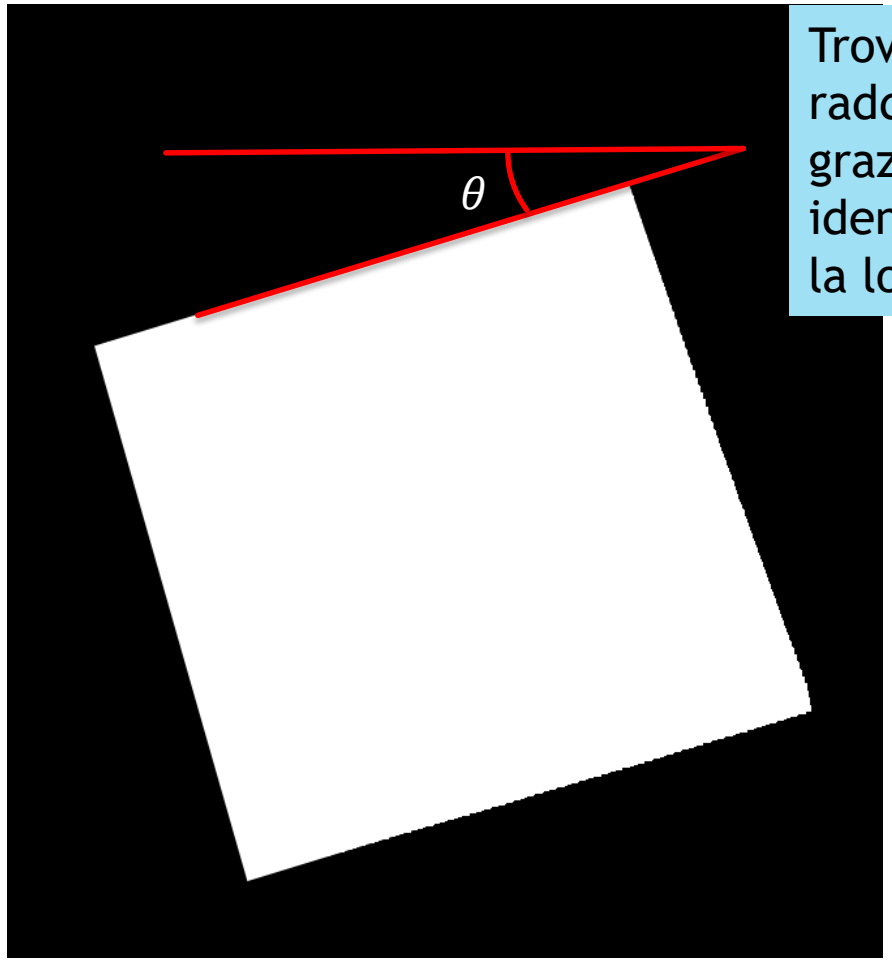


Linee di Hough e punti di intersezione

Figura poligonale inserita

Maschera erosa

Passo 2: correzioni prospettiche

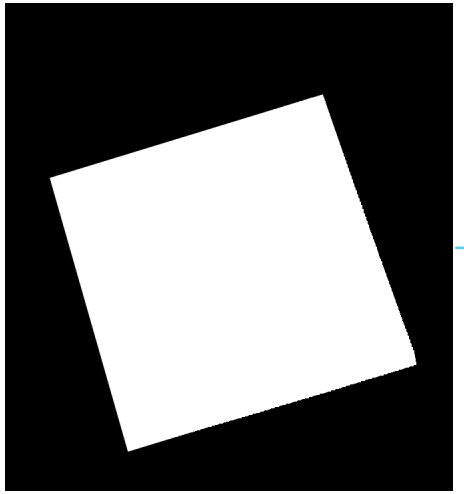


Trovo l’angolo con cui raddrizzare l’immagine grazie alle rette identificate da Hough e la loro inclinazione

Prendo l’immagine originale e le applico tutte le trasformazioni usate finora in modo da rendera sovrapponibile alla maschera trovata

Passo 2: correzioni prospettiche

Maschera



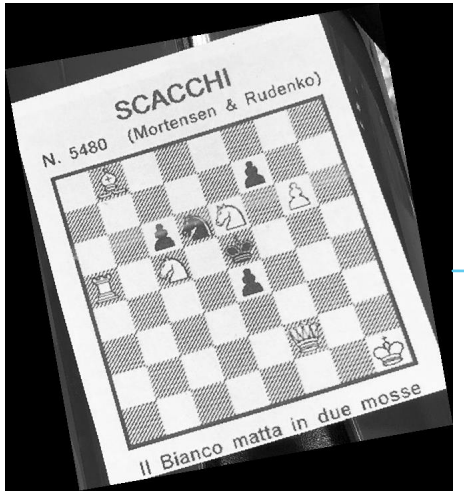
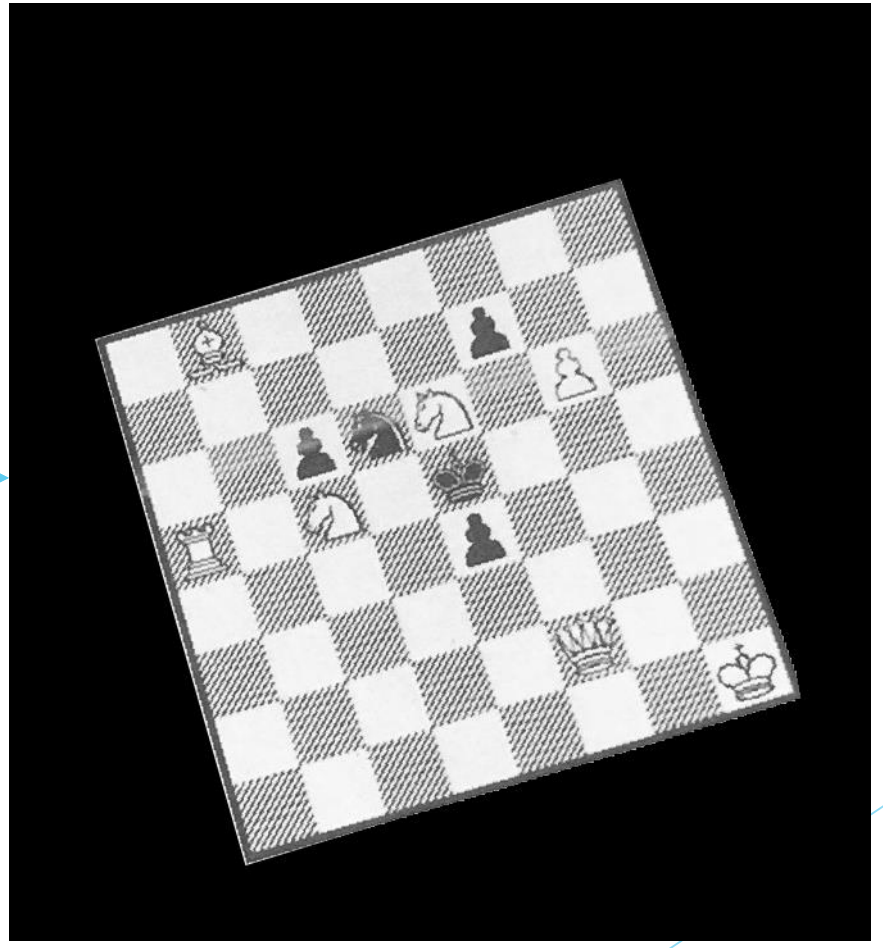


Immagine grayscale della scacchiera

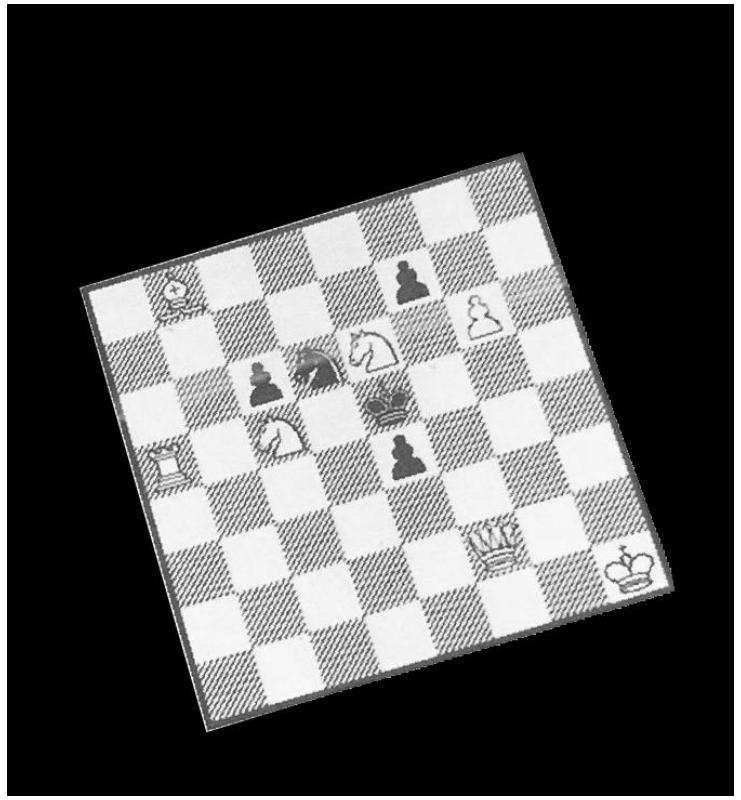
X



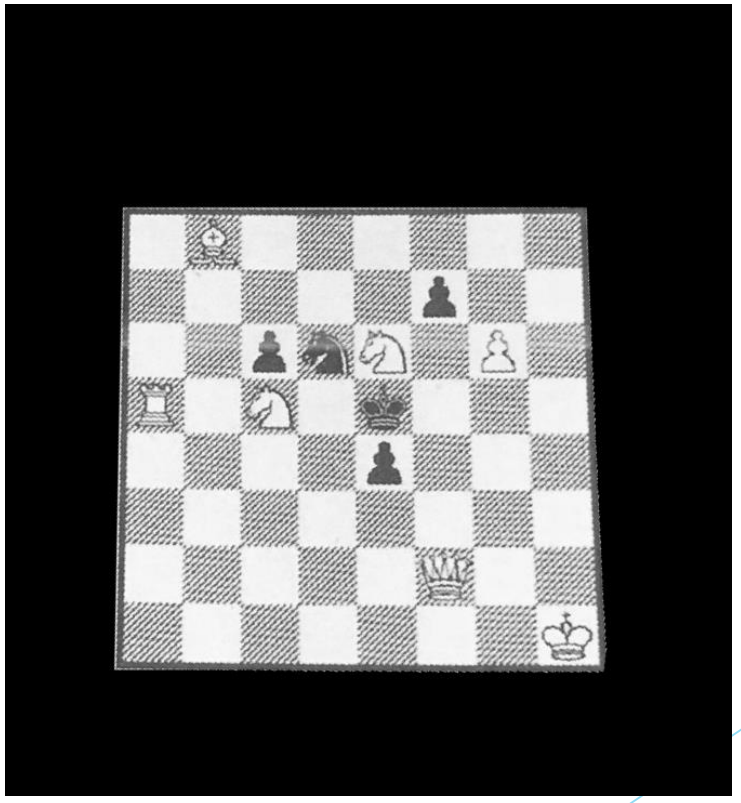
Risultato

Combino la maschera con l'immagine ed ottengo così la sola scacchiera con una buona precisione

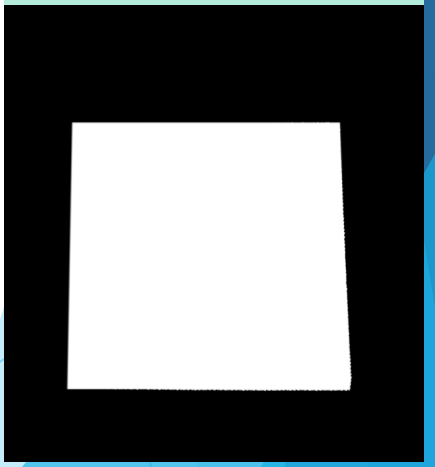
Passo 2: correzioni prospettiche



imrotate

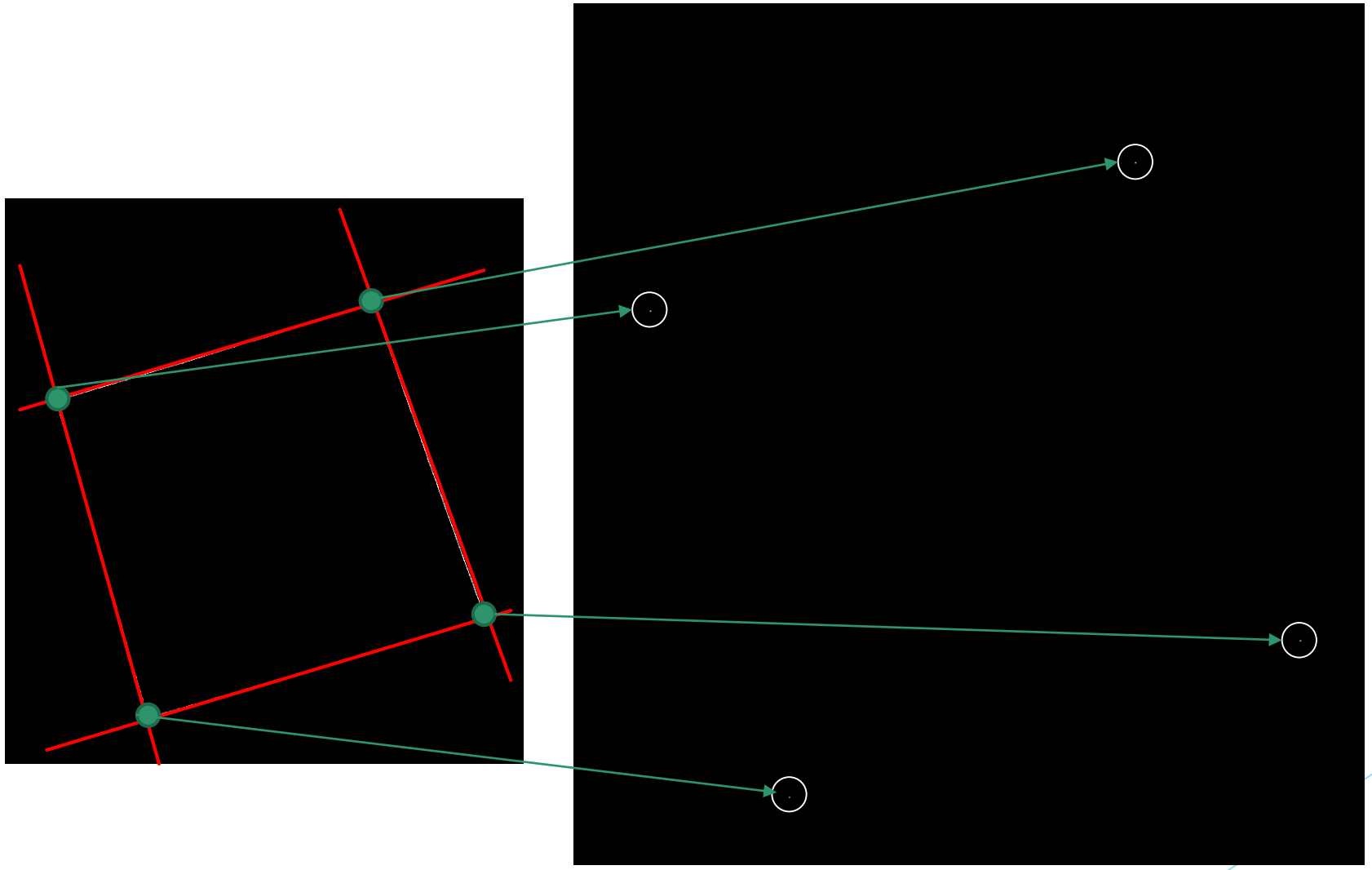


Ruoto anche la maschera per il successivo utilizzo



Ruoto l’immagine con l’angolo trovato in precedenza

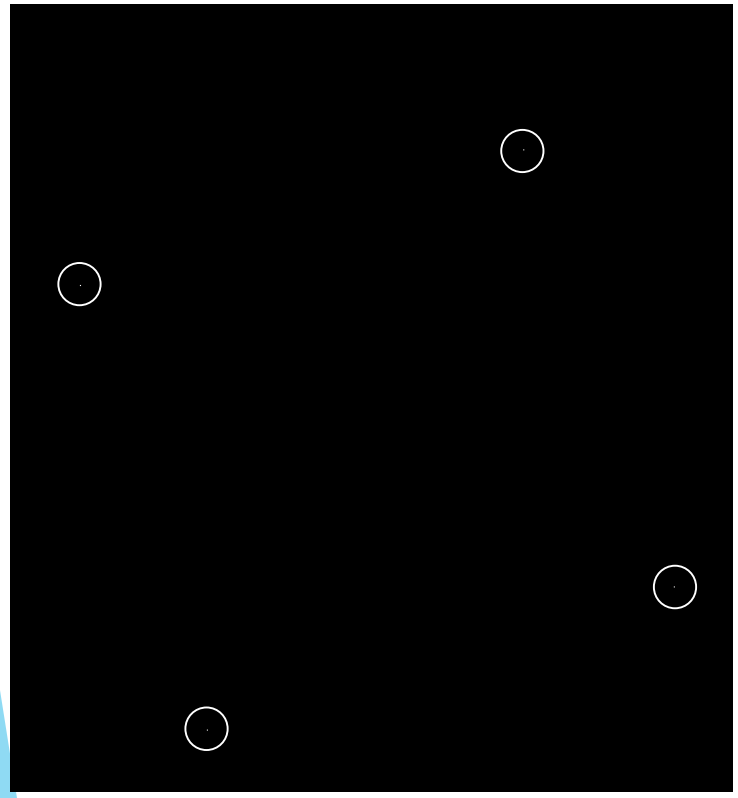
Passo 2: correzioni prospettiche



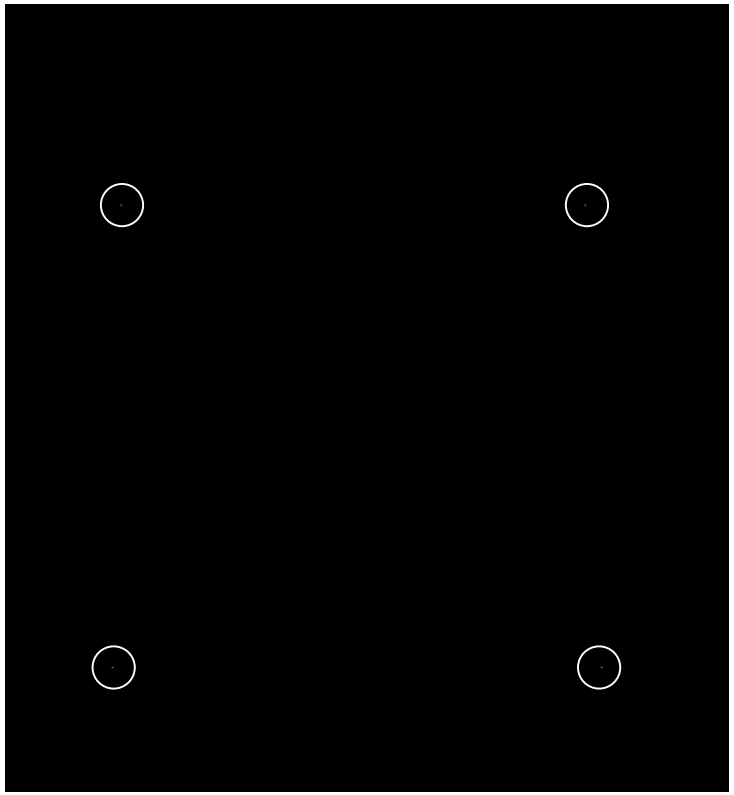
Necessito delle coordinate dei 4 angoli della scacchiera per effettuare la correzione prospettica, Hough ha problemi ad individuare rette troppo verticali per cui utilizzo le coordinate precedentemente trovate sull'immagine non ancora raddrizzata per creare un'immagine delle stesse dimensioni ma contenente solamente 4 punti che corrispondono ai corner della scacchiera.

Passo 2: correzioni prospettiche

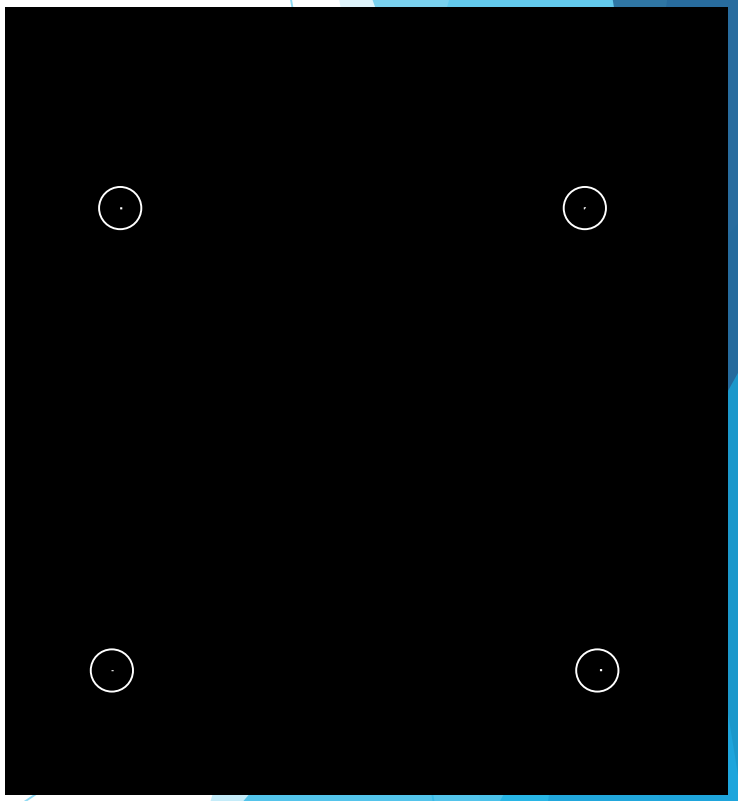
L’immagine viene raddrizzata, questo a causa dell’interpolazione causa la generazione di nuovi punti nell’intorno di quelli originali e rende l’immagine non più binaria ma bensì a livelli di grigio, porto tutto ciò che non è nero al valore 1 tramite sogliatura e riduco il numero di punti a 4 utilizzando il clustering



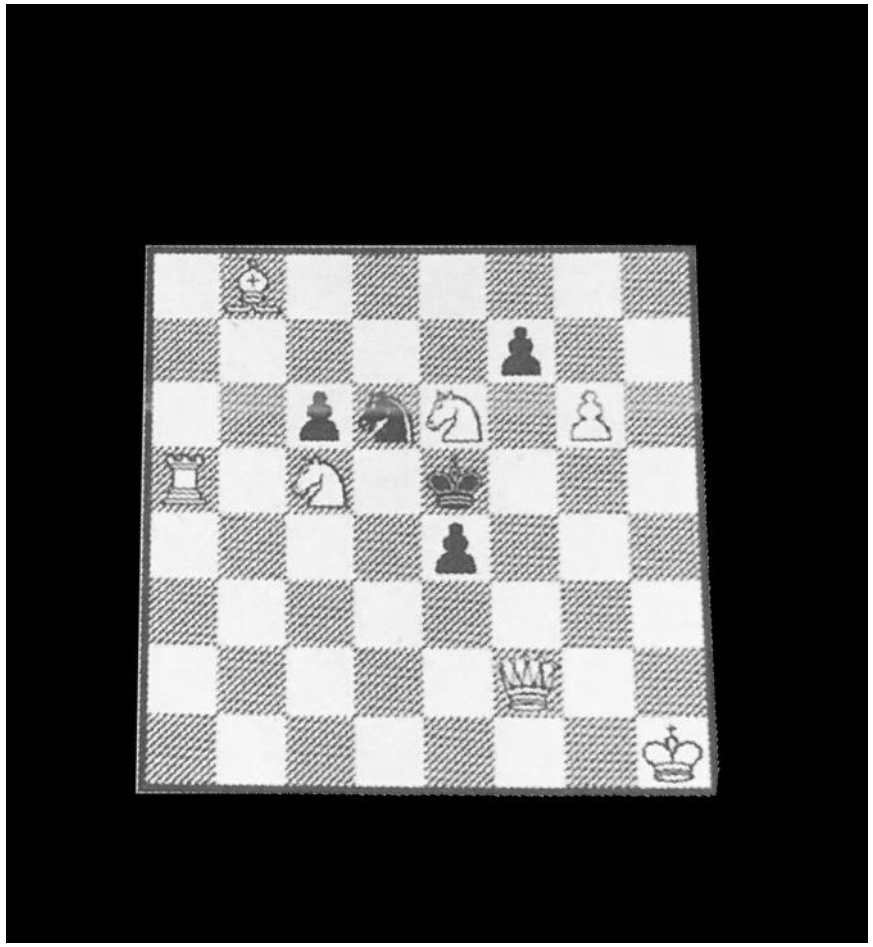
imrotate



sogliatura

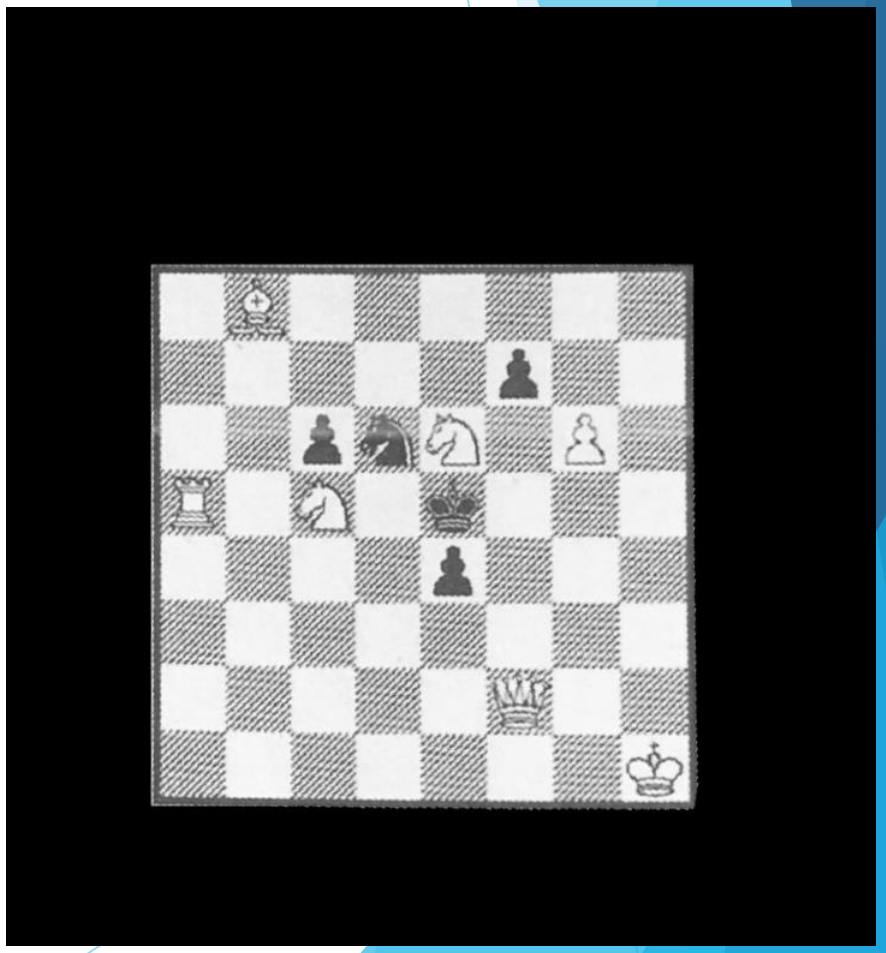


Passo 2: correzioni prospettiche

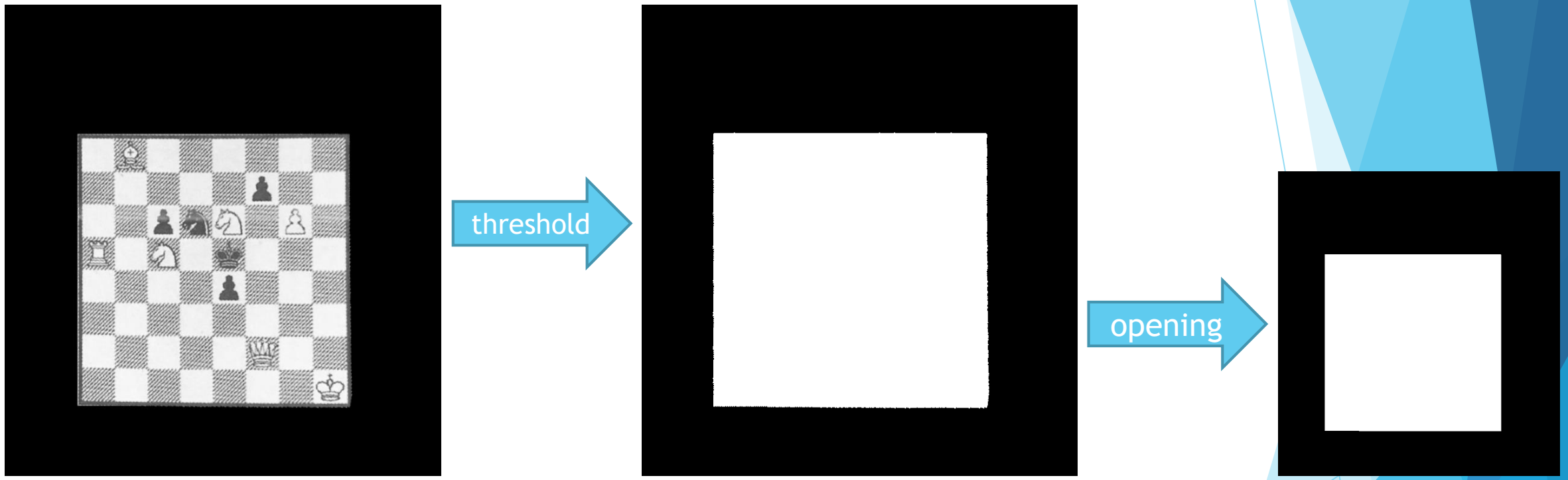


Fitgeotrans + imwarp

Grazie alle coordinate trovate dall'immagine dei 4 punti precedenti applico una trasformazione prospettica per rendere la scacchiera «planare»

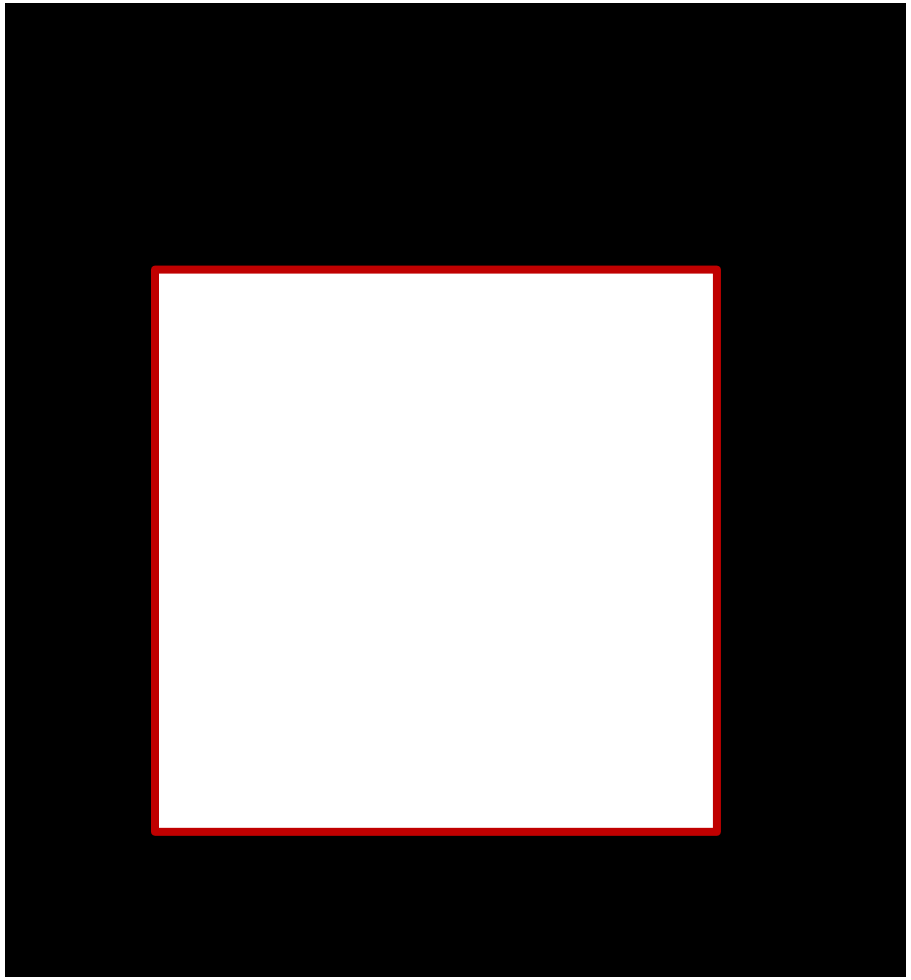


Passo 3: riconoscimento



Soglio l'immagine in modo da trasformarla in una maschera, eseguo 2 opening con elementi strutturanti differenti per perfezionare la maschera

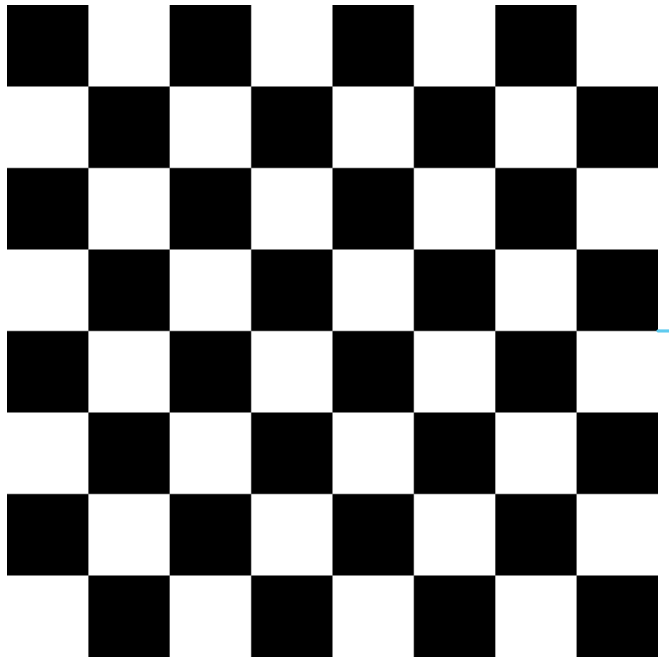
Passo 3: riconoscimento



Trovo la BoundingBox della maschera e ne utilizzo le coordinate per eseguire un crop dell'immagine in modo da ottenere un'immagine occupata interamente dalla maschera, effettuo lo stesso crop con l'immagine della scacchiera.

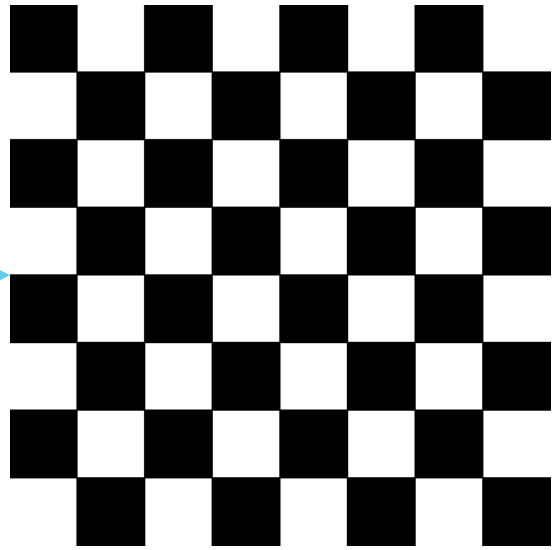
Crop della scacchiera

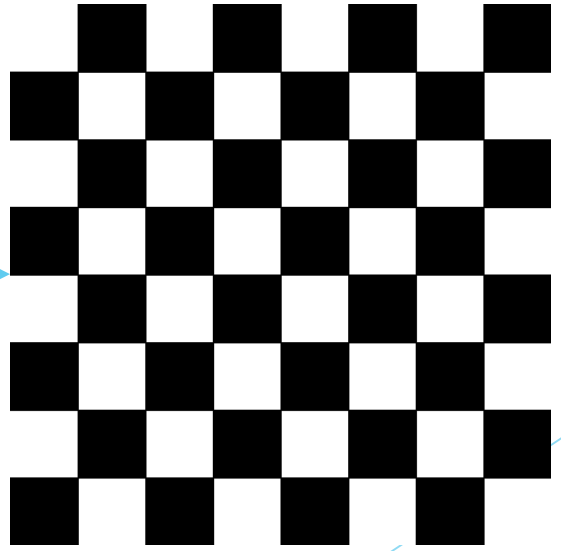
Passo 3: riconoscimento



Checkerboard(8)

Per celle di colore nero





Creo una scacchiera fittizia con circa le dimensioni della scacchiera reale, per i pezzi su cellette nere utilizzo la versione originale, per le cellette bianche invece userò l'inversa.

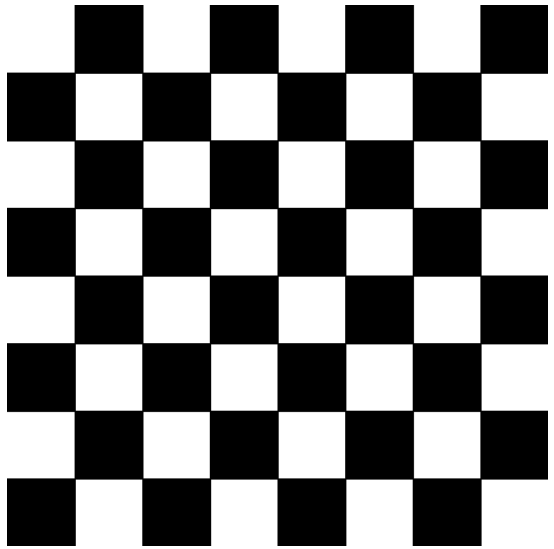
Per rendere ogni cella una componente connessa applico una erosion in modo da separare le celle bianche della scacchiera fittizia.

1 – checkerboard (per le celle di colore bianco

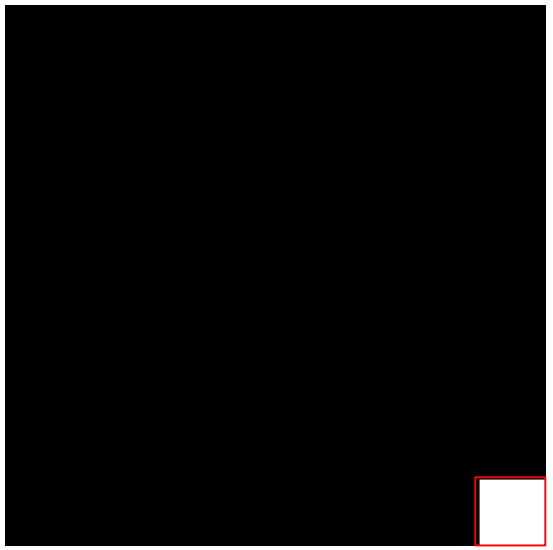
Passo 3: riconoscimento

Effettuando il labelling delle componenti connesse sulle due scacchiere fittizie posso ottenere delle maschere per ogni celletta ed utilizzarle per ottenere la singola cella.

Viene calcolata la BoundingBox sulla maschera della cella e vengono usate le sue coordinate per effettuare un crop sull'immagine in modo da ottenere la singola cella

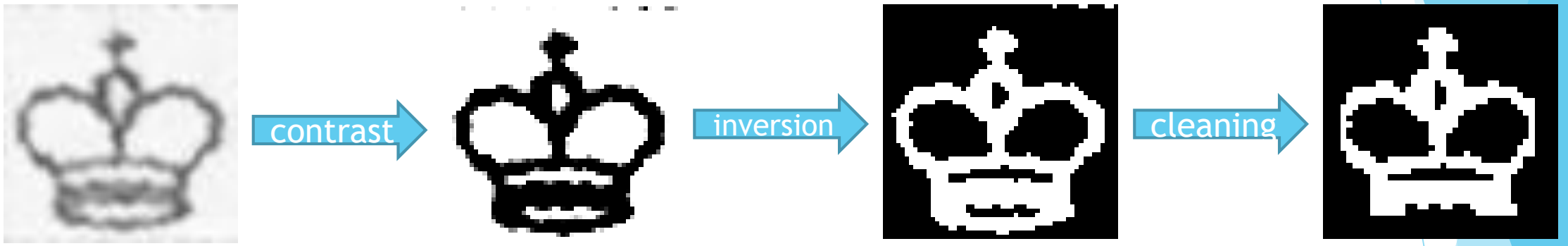


Labelwhite == 32



imcrop

Passo 3: riconoscimento



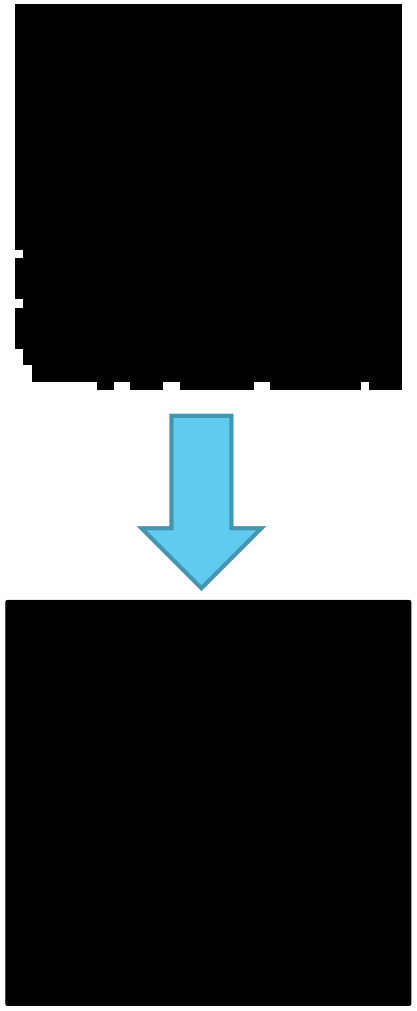
L’immagine della celletta viene prima contrastata in modo da far risultare il pezzo, poi viene binarizzata ed invertita in modo da avere il pezzo con valore 1.
Siccome il crop potrebbe non essere stato perfettamente preciso il bordo interno dell’immagine (di spessore pari a 2px) viene ripulito ponendo tutti i suoi pixel a 0

Un processo del tutto simile viene utilizzato anche per le celle nere.

Passo 3: riconoscimento

Per generare la stringa FEN rappresentativa della scacchiera necessitiamo di poter identificare il «valore» di ogni cella, l'identificazione di celle bianche e nere differisce solo per alcune costanti numeriche e per il metodo riconoscimento delle celle vuote

Mostriamo prima il procedimento per le celle bianche:



Tipica celletta bianca che non contiene alcun pezzo.

L'immagine non appare di colore omogeneo ma contiene imperfezioni dovute alla fase di crop dell'immagine.

Il bordo interno dell'immagine (di spessore pari a 2px) viene ripulito ponendo tutti i suoi pixel a 0

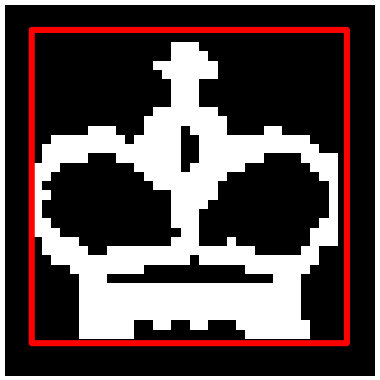
Una cella viene identificata come «vuota» se l'area degli oggetti contenuti al suo interno è inferiore al 5% dell'area dell'intera immagine

D’ora in poi utilizzeremo la parola «valore» (di un’immagine binaria) per indicare il numero di pixel con valore 1 contenuti in essa

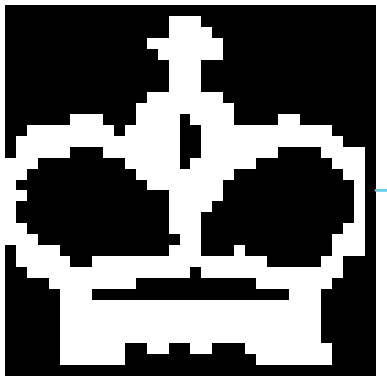
Passo 3: riconoscimento

Se la cella non risulta essere vuota allora il riconoscimento avviene nel seguente modo :

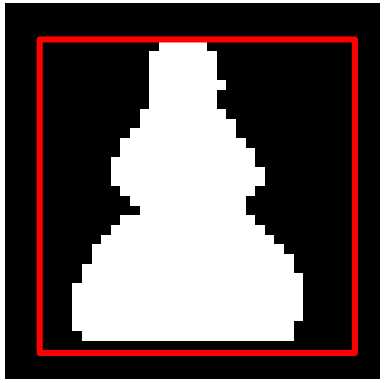
Cella da riconoscere



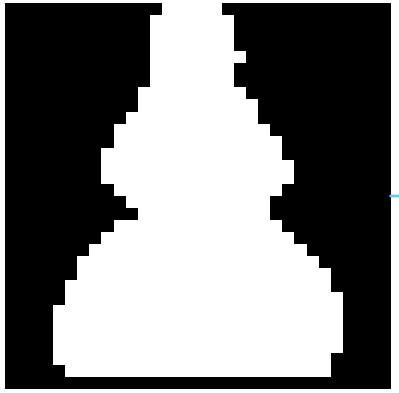
imcrop



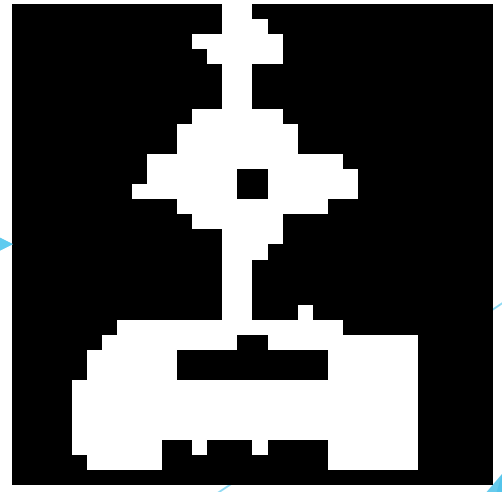
Template



imcrop



X



combinazione

L’immagine della cella ed il template vengono croppati in base alle coordinate della bounding box in modo da centrare meglio l’area di interesse.

Vengono poi combinate, l’immagine risultate viene quindi valutata in base ad alcuni fattori tra cui:

- Valore dell’immagine risultante
- Valore dell’immagine da riconoscere
- Valore del template

I rapporti che esistono tra questi valori generano un valore numerico che viene interpretato come indice di correlazione tra il template e l’immagine.

Passo 3: Riconoscimento

Mostriamo ora il procedimento per le celle nere:

L'immagine viene ripulita da tutte le componenti connesse con area pari ad 1, viene poi ruotata di 45° e croppata utilizzando le coordinate della boundingbox

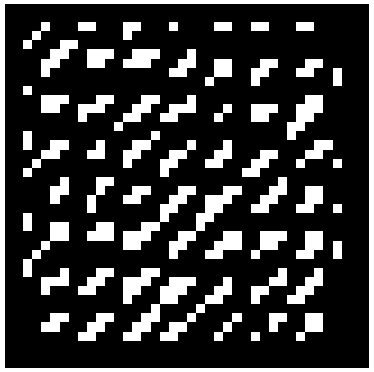
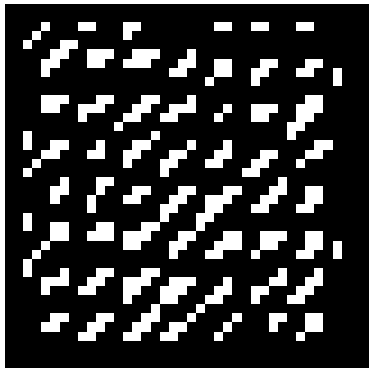


Immagine celletta

labelling



Rimozione pixel singoli

imrotate

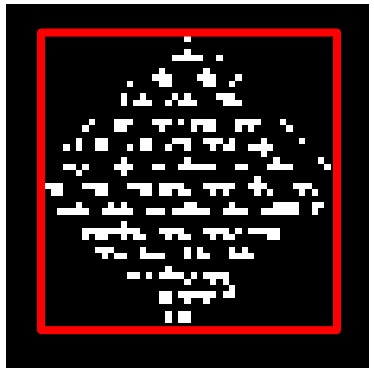


Immagine ruotata

imcrop

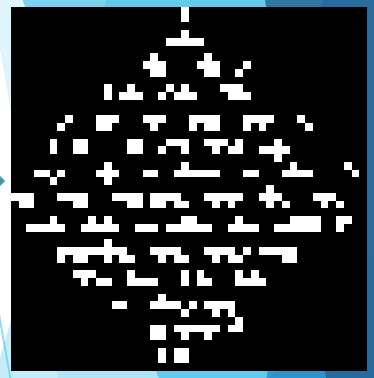
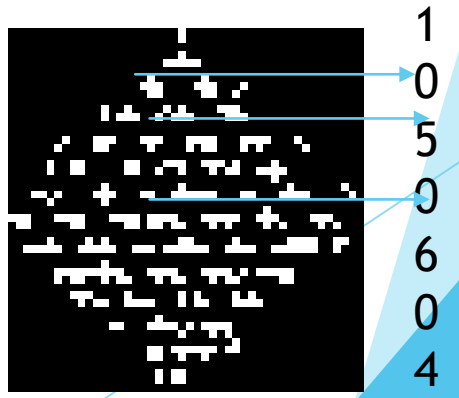


Immagine croppata

L'immagine viene proiettata sommando i valori per ogni riga, se l'immagine rappresenta una cella vuota allora ci dovranno essere molti zeri in questa proiezione.

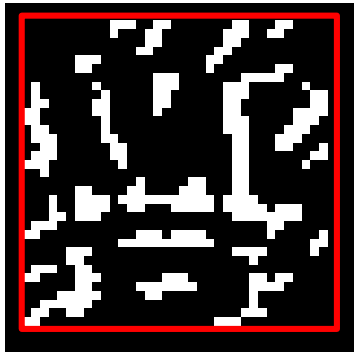
Se il numero di zeri è superiore al 15% del numero di righe totali allora la cella è vuota (non contiene alcun pezzo)



Passo 3: riconoscimento

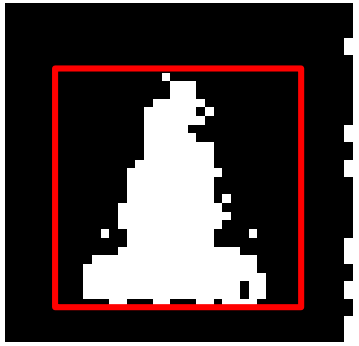
Se la cella non risulta essere vuota allora il riconoscimento avviene nel seguente modo:

Cella da riconoscere

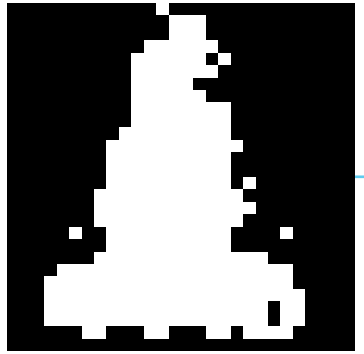


imcrop

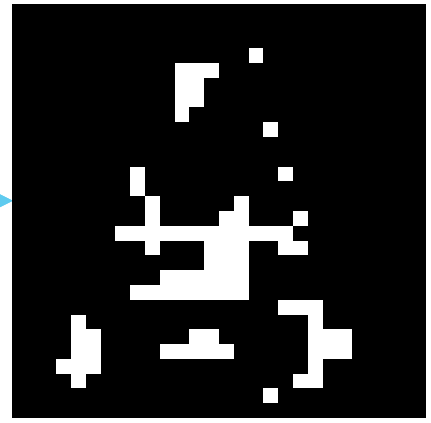
Template



imcrop



X



combinazione

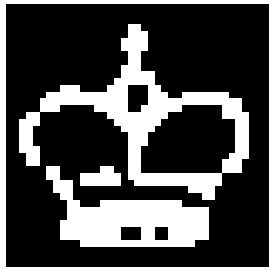
Il procedimento è quasi del tutto simile a quello per le celle bianche, l'unica differenza consiste nel modo in cui si rapportano tra loro i vari valori

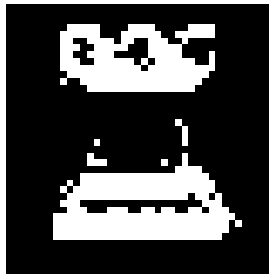
Passo 3: Riconoscimento

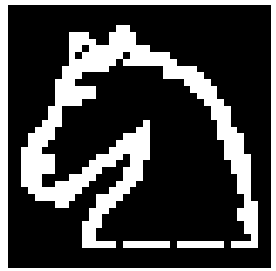
Per il riconoscimento vengono utilizzati dei template, uno per ogni pezzo degli scacchi.
Il template di un pezzo bianco non corrisponde al template dello stesso pezzo nero.
In base al colore della cella vengono usati template differenti.
Le celle bianche hanno un template in più rispetto a quelle nere: la cella vuota.
Non esiste template per il re nero per nessun colore di cella.

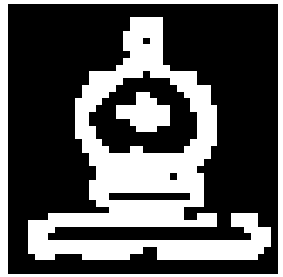
Template celle bianche con rispettivo carattere di codifica FEN:

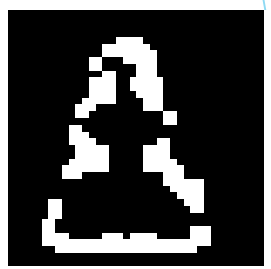

K

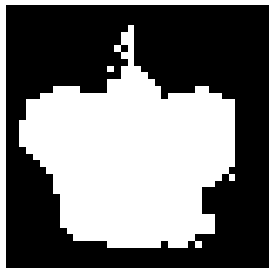

Q

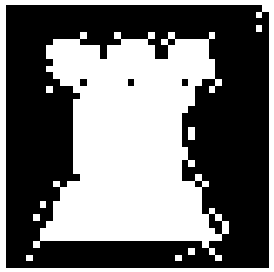

R

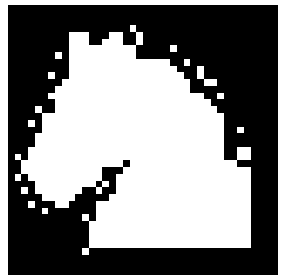

N

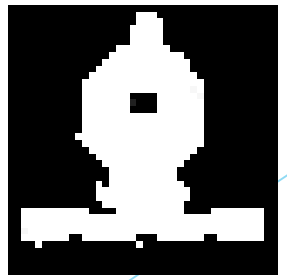

B

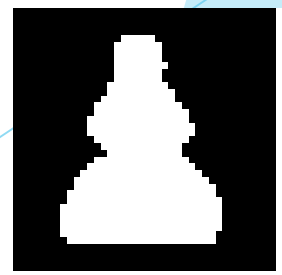

P

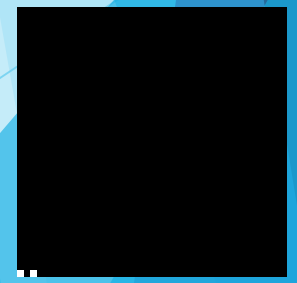

q


r


n


b

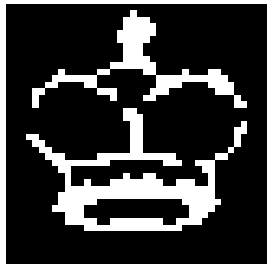

p

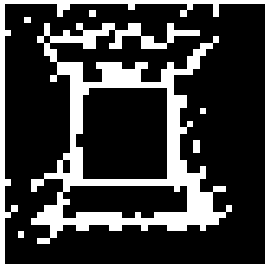

Vuota

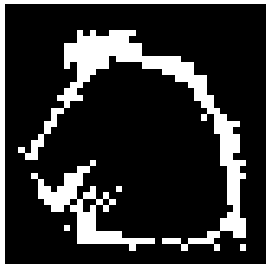
Passo 3: Riconoscimento

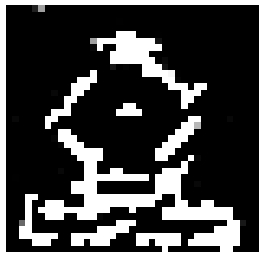
Template celle nere con rispettivo carattere di codifica FEN :

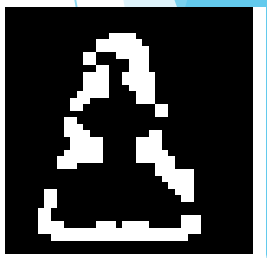

K

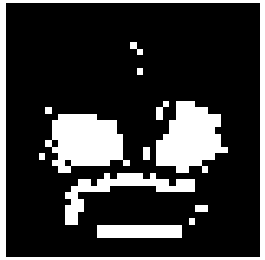

Q

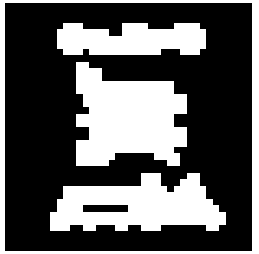

R

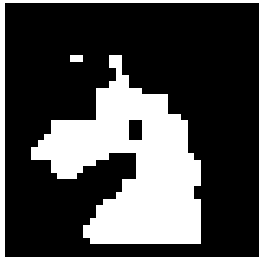

N

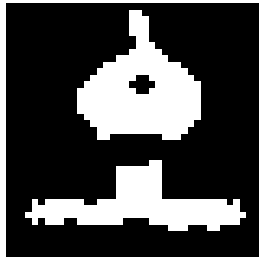

B

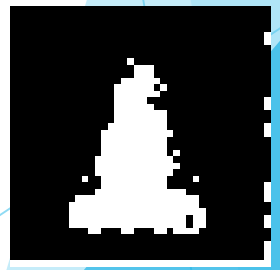

P


q


r


n


b


p

Info Template

I template utilizzati sono immagini binarie.

I template sono stati generati in diversi modi, alcuni sono stati generati partendo dall'immagine di una singola celletta ottenuta tramite lo stesso procedimento usato per il riconoscimento e successivamente perfezionati manualmente tramite editor come Paint ad esempio.

Altri invece sono stati generati mediando alcune immagini di cellette rappresentanti lo stesso pezzo da scacchiere differenti.

Ogni template ha dimensione fissa pari a 40 x 40 px

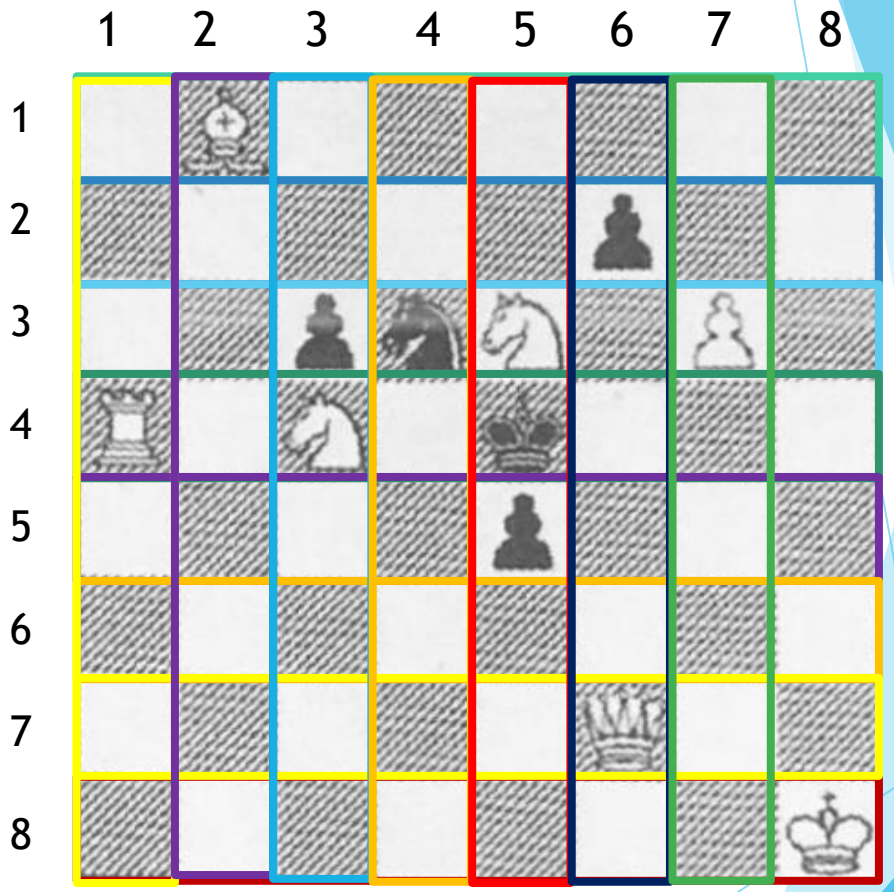
Passo 3: Riconoscimento

La scacchiera viene dunque analizzata riga per riga, per ogni riga vengono estratte 8 immagini rappresentanti le celle della stessa.

Ognuna di queste immagini viene valutata (in base al suo colore conosciuto a priori grazie alla scacchiera fittizia utilizzata) tramite il procedimento mostrato precedentemente con ogni possibile template trovando tra essi la corrispondenza migliore.

In base alla corrispondenza viene dunque creata, riga dopo riga, una matrice 8x8 di valori alfanumerici rappresentanti la codifica di ogni celletta.

Questa matrice viene poi convertita nella stringa FEN vera e propria tramite una funzione apposita.



Parametri e costanti numeriche

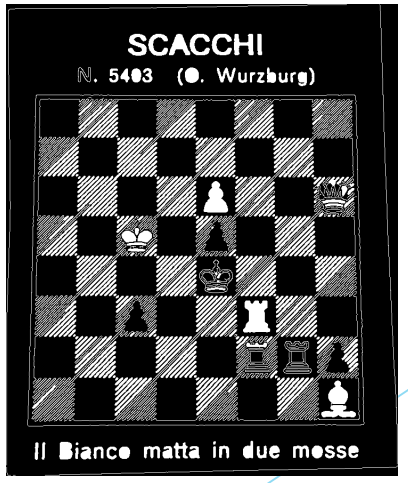
Nell'applicazione vengono utilizzati un numero elevato di costanti numeriche.
Tutte le costanti numeriche hanno origine da un'ipotesi e vengono poi modificate e rese idonee attraverso il «Trial & error».

Ad esempio, per poter distinguere tra una regione rettangolare (foglio intero) ed una regione quadrata (la sola scacchiera) l'ipotesi di partenza corrisponde all'idea che il rapporto l'asse minore e l'asse maggiore della regione ricada in un piccolo intorno del numero 1.
Attraverso il testing ed i suoi risultati questo intorno è stato modificato in modo da generare il minor errore possibile sul test_set portando l'intorno al valore: 0.85 <= Val <= 1.08

L'edge detection tramite il metodo di Canny è stato utilizzato quasi in ogni circostanza con valore di soglia pari a 3.

Il valore è stato ricavato sperimentalmente e le modifiche dello stesso sono ben evidenziate dal successivo utilizzo della funzione imfill()

Soglia = 3



Soglia = 1

Risultati

Immagine	Codifica FEN	Precisione
001	8/8/4p3/2Q1p3/4q3/2p2r2/5RRp/7p - 0 1	97%
002	1B6/5p2/2pnN1Pn/N1N1q3/4pK1n/6n1/5K1q/6pQ - 0 1	89%
003	8/4r1n1/8/R4N2/5K2/3P1N1Q/7p/7B - 0 1	95%
004	R7/1qN1K3/2b5/Rq6/4Np2/8/2R4b/7Q - 0 1	95%
005	7b/8/7R/5R2/8/2QPpB1b/7n/5R2 - 0 1	94%
006	1B1p1B1b/4b1K1/2pKPb1r/K1nNp1b1/1q1RqqKR/pPp1p1p1/1rpb1b1b/q1n1p1b1 - 0 1	56%
007	4rKB1/6n1/3p4/5p2/N2R4/1P6/2q5/QR6 - 0 1	98%
008	2bp1b2/4p3/5q1p/3KNP2/5r1P/8/8/8 - 0 1	94%
009	rp6/nqN1p1r1/1pbp1r1N/PnR1n3/4Np1p/8/1pR4B/4r1rp - 0 1	75%
010	3p4/4r1n1/1K1K1p1p/K1q1nN2/1n3q2/n1RPNNBQ/3p1B1n/4p2R - 0 1	75%
011	1bbb1p2/p1q1p1b1/1p3q1n/p1nKNRp1/1p3R1P/p1p1p1p1/1n1p3q/p1K1b1n1 - 0 1	65%
012	4r1p1/6n1/1n1p4/2n2b2/P2R4/1Pp5/2qp4/pR6 - 0 1	87%
013	1b6/r4p2/1ppnN1P1/r1N1p3/4p3/8/5K2/7p - 0 1	90%
014	8/8/4p2K/2Q1p3/4q3/5r2/5RRp/7b - 0 1	97%
015	1R6/8/2K1P3/3Np3/4q1Bp/1P6/2Q5/2n3b1 - 0 1	95%
016	8/7n/7B/5p2/8/2bP1p1p/8/4Rb2 - 0 1	86%
017	1n6/5p1R/2ppq1p1/B1N1pR1R/4p1R1/n7/5R2/7p - 0 1	79%
018	Non riconosciuta	0%
019	R7/1qN1r3/2b5/1n6/4Np2/8/2R4R/7Q - 0 1	95%
020	4rKp1/6n1/3p4/5p2/N2K4/1p5K/2n4b/1p6 - 0 1	86%
021	n7/b3r1n1/8/Q4R2/5pn1/3p1R1K/7n/7p - 0 1	84%
022	Non riconosciuta	0%
023	2bb1Q2/4p3/5p1n/3pNP2/5P1P/8/8/8 - 0 1	95%
024	1B6/8/1rK1P3/3Np3/1q1nq1BP/1P6/1rQ5/2n3b1 - 0 1	90%

Risultati

Immagine	Codifica FEN	Precisione
025	Non riconosciuta	0%
026	8/8/4p2p/2b1p3/4q3/2p2r2/5RRp/7b - 0 1	95%
027	8/5p2/2ppN1P1/R1N1q3/4p3/8/5Q2/7K - 0 1	94%
028	4rpKn/r1n1n1n1/1r1p3n/5bn1/NN1R1N1r/1PR1n1n1/1rqn1N1n/pbN1p1n1 - 0 1	62%
029	2bp4/4p3/7p/3KNP2/5r1P/8/8/8 - 0 1	92%
030	1b6/8/2K1P3/3Np3/4q1bN/1P6/2Q5/2p5 - 0 1	90%
031	r7/pqN1Q3/2b5/Pn4R1/4Np2/8/2R4p/7p - 0 1	90%
032	8/8/8/5R2/8/3PqB1b/8/5R2 - 0 1	95%
033	8/4r1n1/8/K4N2/5q2/3P1N1b/7p/7b - 0 1	95%
034	8/8/4p2R/2b1p3/4q3/2p2r2/5NRp/7b - 0 1	94%
035	4rKb1/6n1/3p4/5p2/N2R4/1P6/2q5/QR6 - 0 1	97%
036	2bp1b2/4p3/5q1n/3KNB2/5p2/8/8/8 - 0 1	92%
037	Non riconosciuta	0%
038	8/8/8/5R2/8/2bPqb1b/8/5R2 - 0 1	94%
039	R7/1qN1K3/2b5/Pn6/4Np2/8/2R4B/7K - 0 1	100%
040	Non riconosciuta	0%
041	8/4r1n1/8/B4N2/5K2/3p1R1p/7n/7p - 0 1	90%
042	R7/1qN1r3/2pb4/1p6/4rp2/8/2B4b/7K - 0 1	86%
043	1B6/5p2/2ppN1P1/N1N1q3/4p3/8/5K2/7Q - 0 1	97%
044	2bb1b2/4p3/7n/3pNP2/5P1R/8/8/8 - 0 1	92%
045	8/8/8/5R2/8/3PqB1b/8/5R2 - 0 1	98%
046	4bpp1/6n1/3p4/5b2/N2P4/1P6/2q1n3/pp1n1R1R - 0 1	84%
047	1b6/8/2p1N3/3Pb3/4q1RR/1P6/2p5/2n4R - 0 1	86%
048	8/8/4p2K/2Q1p3/4q3/2p2r2/5rrp/7p - 0 1	95%

Analisi dei risultati e delle performance

Il Dataset è costituito da 48 immagini a colori generate da dispositivi differenti in condizioni differenti.
N.B. Alcune immagini sono state ruotate in modo tale per cui la scacchiera avesse un angolo compreso tra i -45 e i 45 gradi rispetto all'asse delle ascisse.

Di queste 48 immagini:
- 43 vengono riconosciute con una precisione dell' **89%**
- 5 non vengono riconosciute a causa dei seguenti problemi:

Assegnando una precisione pari allo 0% alle immagini che non vengono riconosciute, la precisione relativa all'intero dataset è pari all' **80%**

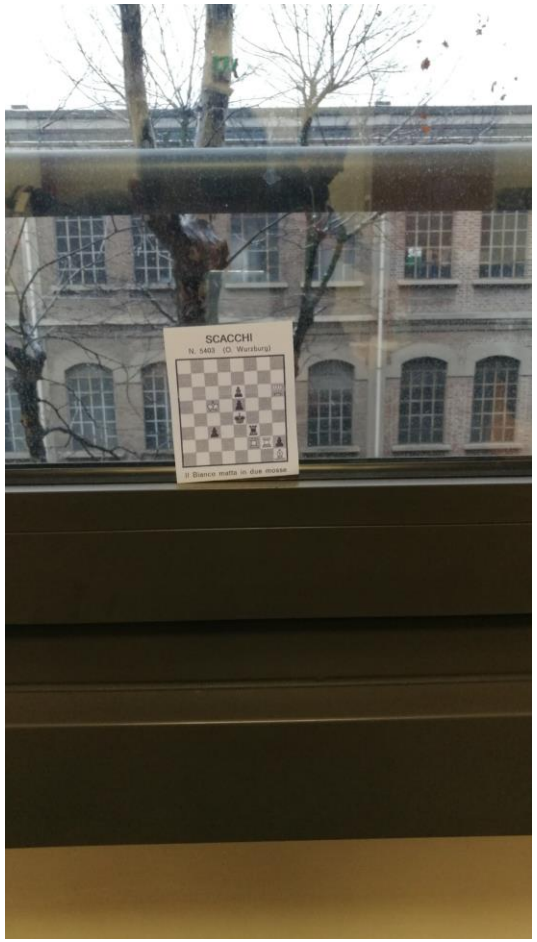
Img 18

Img 40

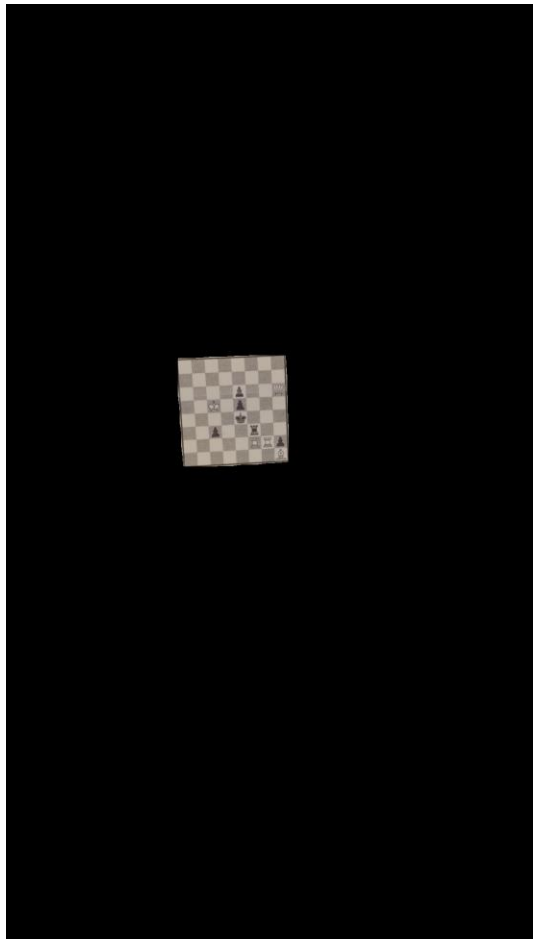
Le immagini numero 18 e 40 non vengono riconosciute in quanto la linea nera che forma un quadrato che circoscrive la scacchiera non è continua a causa di un riflesso (18) o per la presenza di un dito (40).

Il problema in questo caso appare in fase di individuazione della scacchiera, la linea non continua genera un edge non continuo rendendo l'utilizzo di imfill() inefficace.

Analisi dei risultati e delle performance

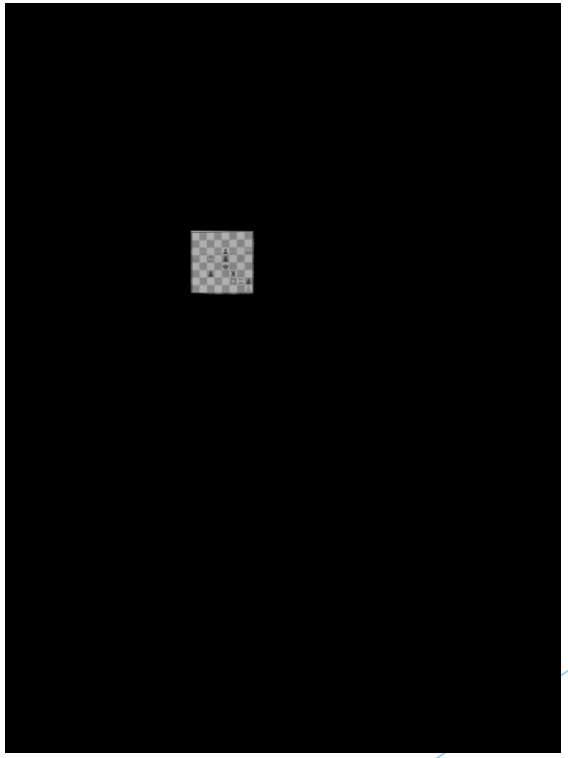


Img 22



Passo 1

NOTA: Nel caso un cui venisse effettuato un crop maggiore del 10% dell'immagine 22 essa verrebbe riconosciuta con una precisione dell'87,5%



Passo 2

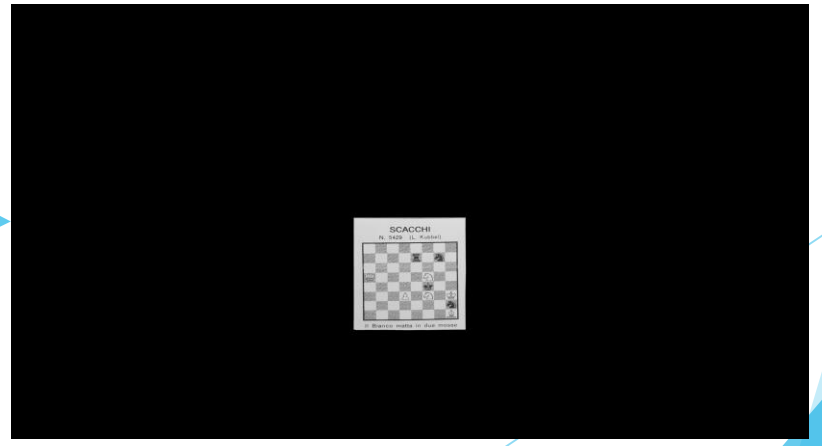
L'immagine numero 22 supera brillantemente il passo di identificazione della scacchiera e la sua correzione prospettica ma fallisce in seguito a causa del rapporto troppo piccolo tra dimensione della scacchiera e dimensione dell'intera immagine

Analisi dei risultati e delle performance

Img 25

L'immagine numero 25 supera tranquillamente il primo passo di identificazione della scacchiera ma a causa della prospettiva con cui è stata scattata l'immagine l'asse maggiore e l'asse minore del foglio hanno dimensioni troppo simili (il foglio è più simile ad un quadrato che ad un rettangolo) e questo fa si che nel passo 2 l'intero foglio venga confuso per la sola scacchiera.

Risultato passo 1

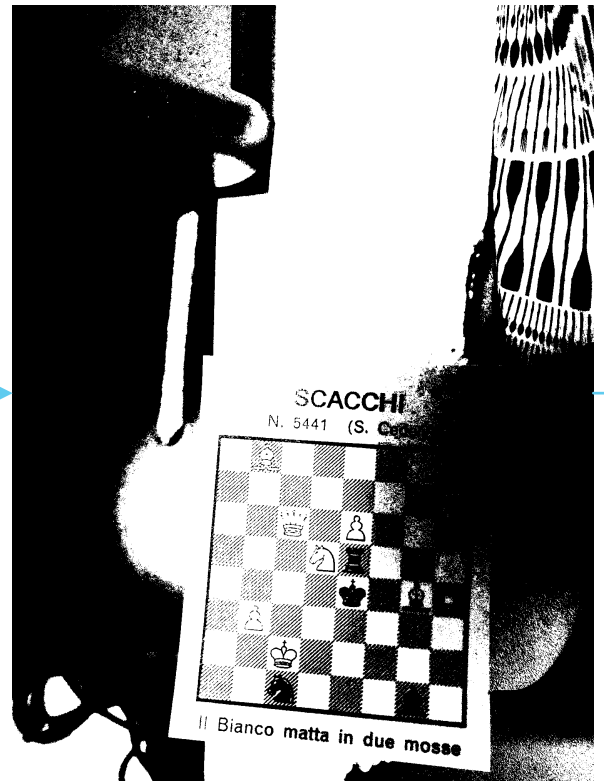


Risultato passo 2

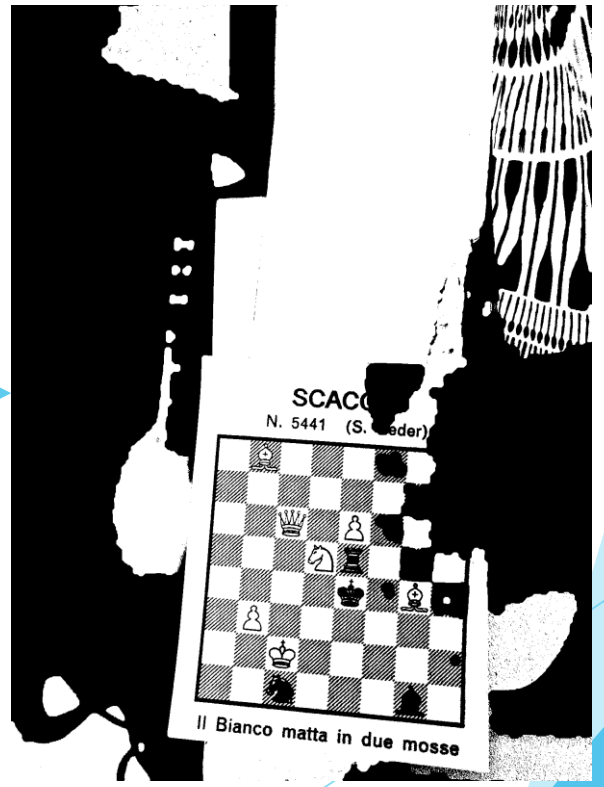
Analisi dei risultati e delle performance

L’immagine numero 37 non supera il primo passo di identificazione a causa dell’ombra troppo netta proiettata sulla scacchiera.

Img 37



Canale Value



Combinazione del canale V con le altre maschere

In particolare il canale Value dello spazio colore HSV, nonostante la gamma correction precedente, non riesce ad evidenziare correttamente la scacchiera e pregiudicando dunque la formazione di una maschera precisa

Analisi dei risultati e delle Performance

Analizziamo ora i dati relativi al riconoscimento dei singoli pezzi tramite la matrice di confusione calcolata sulle 43 immagini riconosciute

	vuota	K	Q	q	R	r	N	n	B	b	P	p
vuota	2398	15	23	10	11	9	12	8	21	5	25	13
K	2	25	3	0	2	2	0	0	1	0	0	1
Q	4	3	21	0	0	0	0	0	0	8	0	11
q	2	2	0	39	0	0	0	1	0	0	0	4
R	2	1	0	0	39	6	4	0	3	3	4	4
r	0	0	0	0	0	18	0	0	0	2	0	4
N	0	0	0	1	3	1	47	0	0	0	2	0
n	0	0	0	1	0	0	0	32	0	0	0	9
B	1	2	0	0	4	0	1	0	20	11	0	9
b	2	0	0	0	0	0	0	0	0	26	0	8
P	3	0	0	0	2	0	1	0	1	0	43	3
p	1	0	0	0	0	0	0	0	0	0	0	59

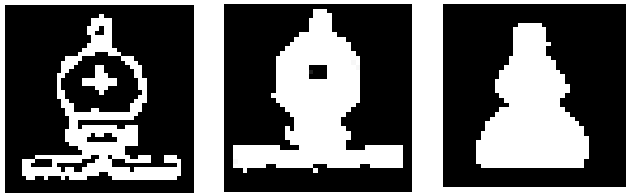
Analisi dei risultati e delle Performance

Dalla matrice possiamo effettuare le seguenti osservazioni:

vuota	K	Q	q	R	r	N	n	B	b	P	p
94%	58%	47%	81%	59%	75%	87%	76%	41%	72%	79%	98%

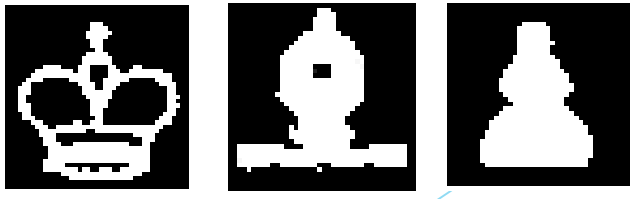
Precisione del riconoscimento del singolo pezzo espresso in percentuale

Il pezzo più problematico per il riconoscimento è evidentemente l'alfiere bianco, la matrice di confusione precedente fornisce dati sui quali si può ipotizzare che la motivazione alla base di questo problema è l'eccessiva somiglianza dell'alfiere bianco con la sua controparte nera ed il pedone nero.



B b p

Al secondo posto come performance peggiore troviamo la regina bianca che in base alla matrice di confusione precedente pare venir confusa principalmente con l'alfiere nero ed il pedone nero. Si può ipotizzare che ciò sia dovuto alla densità della figura della regina nella sua zona centrale che ben corrisponde alle altre due figure.



Q b p

Analisi dei risultati e delle performance

La maledizione del pedone

In generale è possibile notare come un numero sopra la media di pezzi venga erroneamente scambiato per il pedone nero, infatti su 304 errori di riconoscimento, 66 volte il pezzo venga confuso con il pedone nero (22% degli errori contro un 8% teorico ipotizzando che la probabilità sia la stessa per ogni pezzo).

Di contro ben il 98% dei pedoni neri vengono correttamente riconosciuti come tali.

La scacchiera che ottiene la precisione peggiore è la numero 6 con una precisione di appena il 56%

La motivazione la si può cercare nel fatto che l'immagine soffra dell'effetto mosso a causa del quale le celle vuote di colore nero non vengono correttamente identificate dal criterio scelto

La scacchiera che invece ottiene il risultato migliore con una precisione del riconoscimento del 100% è la numero 39