ELABORAZIONE DELLE IMMAGINI
Anno di corso 2013-2014

RICONOSCIMENTO AUTOMATICO DI OGGETTI D'INTERESSE E CIBI SU UN VASSOIO

Alessandro Leonetti # 746215
Riccardo Cassataro # 736424

Obiettivi dell'algoritmo

- Data in input un’immagine l'algoritmo dovrà localizzare:
- 1) Bicchieri (distinguendo tra capovolti e non);
- 2) Contenitore dei mandarini;
- 3) Piatti piani e fondi circolari di dimensioni varie;
- 4) Purè di patate / polenta;

Premesse

- L'algoritmo è stato implementato in Matlab data la sua buona capacità di operare su matrici ed i numerosi tool offerti;
- Si presume che la scena sia illuminata sempre dalla stessa luce (vale quindi per tutte le immagini del training set);
- Le immagini vengono scattate da una camera fissa, quindi gli oggetti avranno sempre più o meno le stesse dimensioni e scale;
- I mandarini sono sempre in coppia e sono contenuti in un contenitore bianco, rettangolare, di uguali dimensioni;
- In ogni immagine possiamo trovare:
  - 1 sola scatola di mandarini;
  - 0 o più bicchieri;
  - 0 o più piatti circolari di varie dimensioni.

Passi Principali

- Prendere in input un'immagine dal dataset;
- Effettuare del Preprocessing applicando il White Point Balance ed eliminando eventuali dominanti colore dall'immagine originale;
- Applicare le maschere e gli algoritmi costruiti per ogni elemento;
- Riconoscere e localizzare gli elementi d'interesse.

White Point Balance
Un po di teoria..

- Aiuta a rimuovere o correggere la dominante colore dovuta al tipo d’illuminazione della scena ritratta nell'immagine.
- Il valore massimo Rmax, Gmax, Bmax di ogni canale R, G, B viene fatto corrispondere ad un valore massimo di riferimento WhiteR, WhiteG, WhiteB. L'algoritmo cerca il punto bianco nell'immagine:

  - kR = WhiteR/Rmax
  - kG = WhiteG/Gmax
  - kB = WhiteB/Bmax

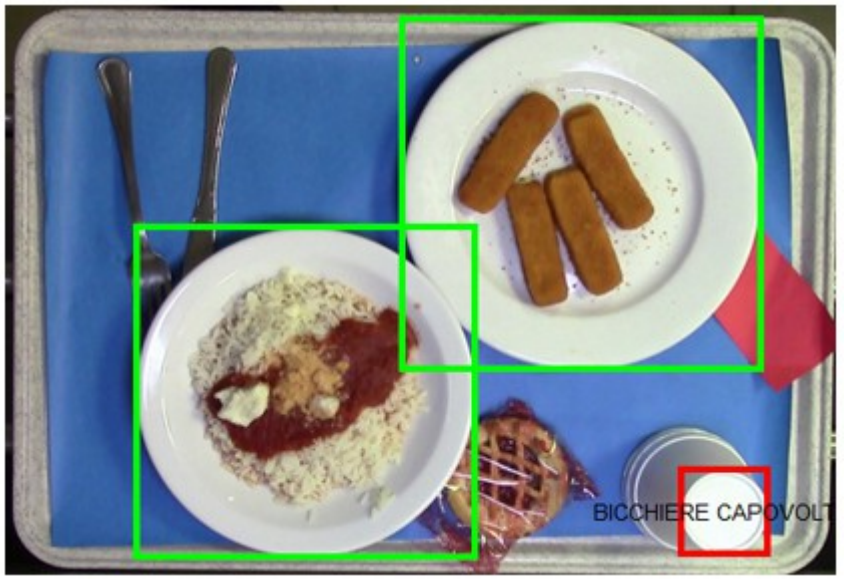
- Nel nostro caso il bianco di riferimento per la tripla (R, G, B) è (255, 255, 255). Applicando l'algoritmo è possibile osservare un lieve miglioramento nelle immagini, che tuttavia influisce in modo praticamente impercettibile sul riconoscimento degli oggetti d'interesse. Dimostrazione in fase di colloquio.

Riconoscimento Bicchieri

- Trasformata di Hough: E' una tecnica che consente di analizzare la collinearità dei punti di edge in modo efficiente. Viene impiegata soprattutto per individuare la presenza di particolari curve parametriche (Es. Circonferenze, ellissi o parabole). È un metodo ad accumulo di evidenza nel senso che ogni pixel contribuisce ad indicare delle curve alle quali appartiene;
- Grandi valori rappresentano linee significative, mentre i valori più piccoli corrispondono o a punti isolati o a segmenti corti (spesso rimovibili).

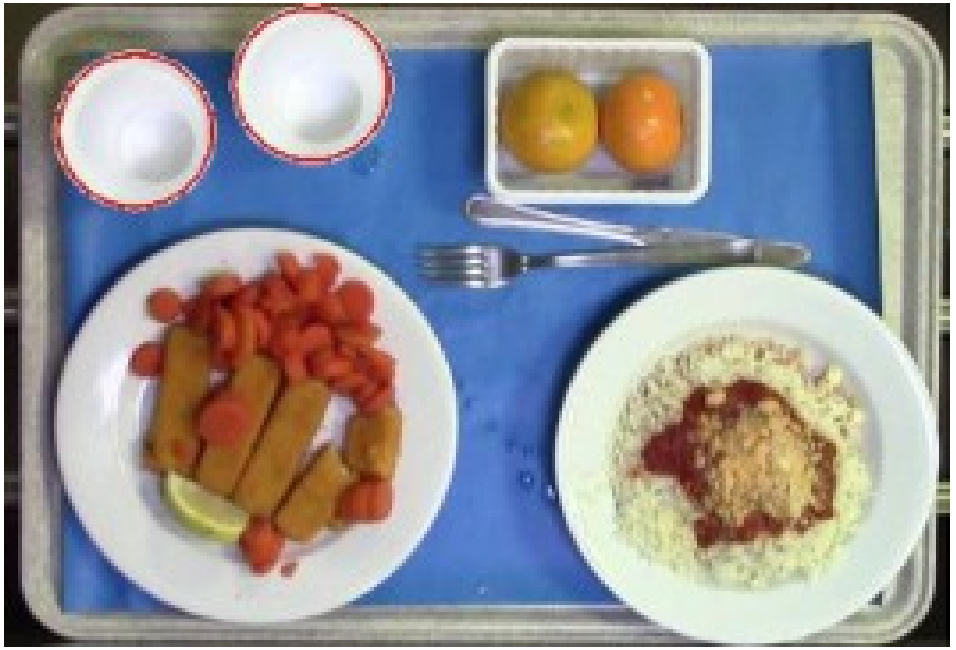
Punti Principali del Funzionamento

- Prendere in input l'immagine del dataset preprocessata;
- Determinare il raggio dei bicchieri in piedi e di quelli capovolti (per tentativi);
- Richiamo della funzione "ImFindCircles" per discriminare gli oggetti, essa restituisce in output un vettore contenente le coordinate del centro delle circonferenze trovate e la misura dei loro raggi;
- Se il raggio è compreso tra 20 e 30 verrà identificato un bicchiere capovolto (in alcuni casi il vasetto dello yogurt come falso positivo), se invece è compreso tra 35 e 50 verrà identificato il bicchiere in piedi;



Passo Intermedio
Funzione: ImFindCircles

L'immagine sottostante è un risultato intermedio ottenuto con l'applicazione della funzione ImFindCircles. Come descritto nella slide precedente essa prende in input un'immagine ed un vettore contenente i valori dei raggi d'interesse da cercare nell'immagine. Restituisce in output un vettore contenente le coordinate del centro delle circonferenze trovate e la misura dei loro raggi.



Riconoscimento Piatti

- Per il riconoscimento dei piatti abbiamo usato esattamente lo stesso procedimento dei bicchieri, ma usando un range differente e più ampio per cercare nell'immagine delle circonferenze più grandi. In basso è riportato il risultato della computazione dell'algoritmo sull'immagine "148.png".

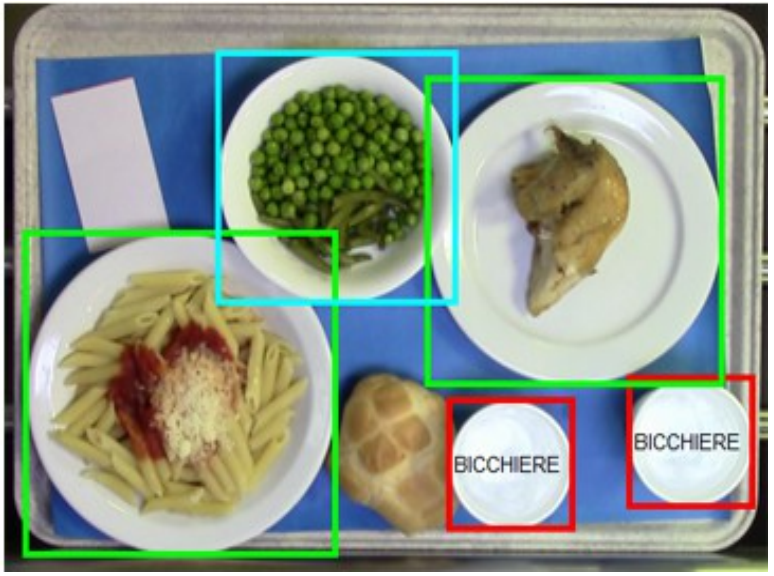


Tabella riassuntiva dei parametri utilizzati

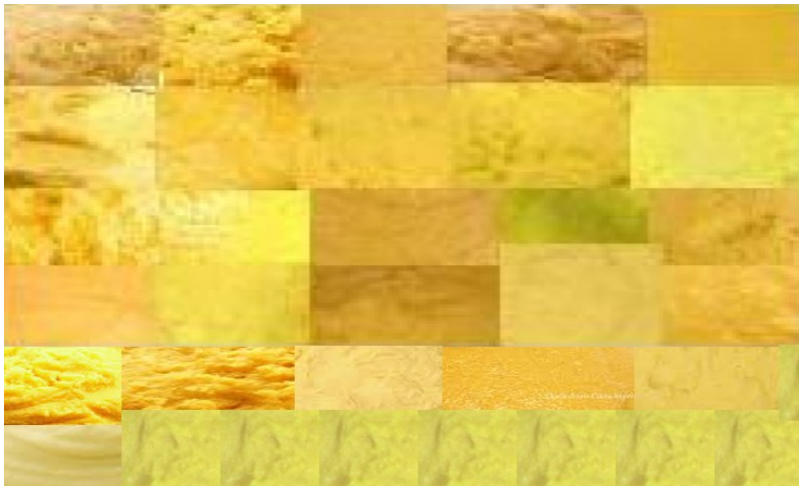
- Nella tabella sottostante abbiamo raccolto i parametri utilizzati dalla funzione "imfindcircles" per riconoscere le circonferenze richieste.

Oggetto Riconosciuto	**Raggio min**	**Raggio MAX**
Bicchiere	35	50
Bicchiere Capovolto	20	30
Piatto Piano	85	150
Piatto Fondo	50	80

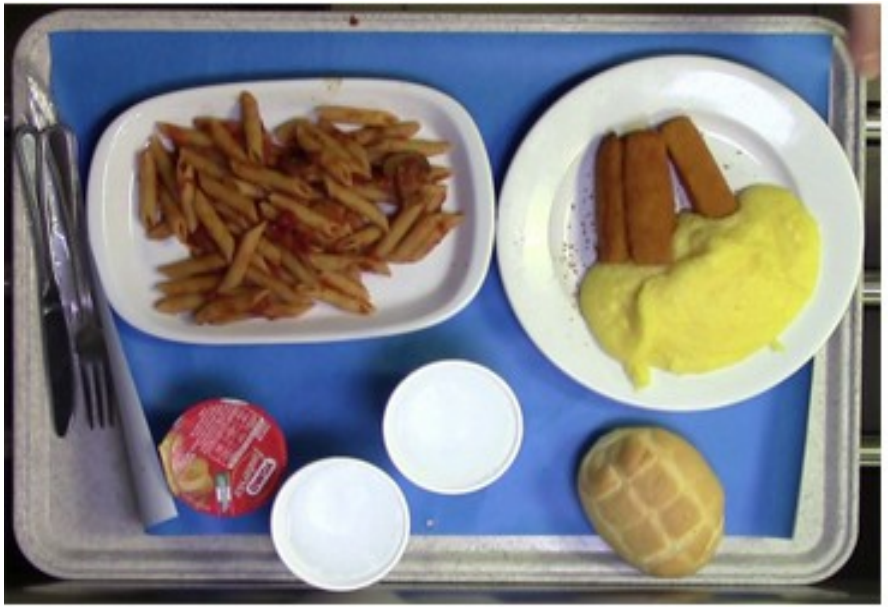
Passi per il riconoscimento del Purè e della Polenta

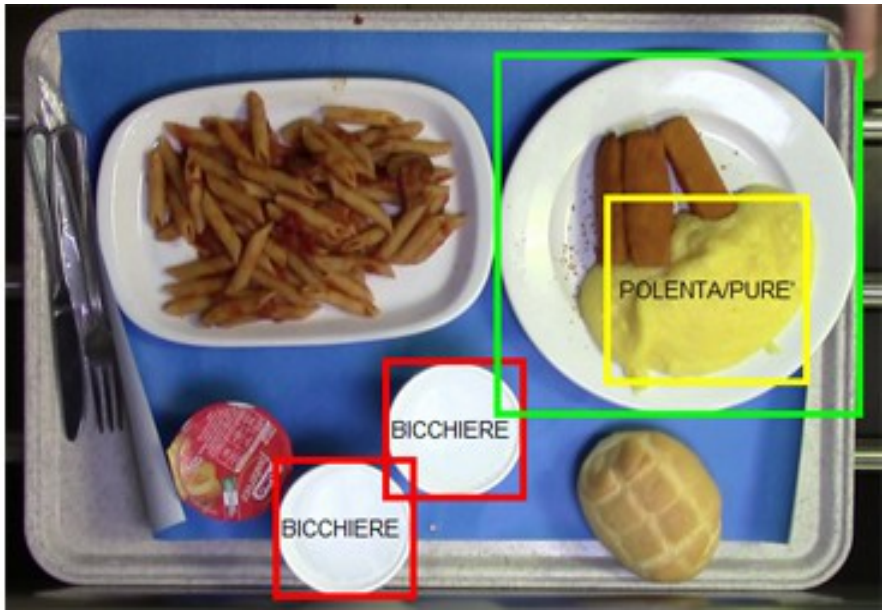
- L'algoritmo prende in input un'immagine di skin grazie alla quale potrà riconoscere i campioni di purè e polenta ed un'immagine del dataset;
- Entrambe vengono trasformate dallo spazio RGB allo spazio HSV e vengono calcolate le proprietà dell'immagine ed estratte Hue, Saturation e Value;
- Creazione di una maschera che riconosce ed evidenzia i candidati positivi tramite il riempimento di tre matrici (una per ogni canale HSV) di booleani per determinare i pixel candidati;
- Calcolo delle componenti connesse per espandere le regioni;
- Applicazione di un filtro tramite bwareaopen(pixel 1200, connettività 8), per eliminare gli eventuali elementi corrispondenti ai falsi negativi individuati dalla maschera;
- Visualizzazione dell'output;

Immagini di funzionamento



L'immagine a sinistra è l'immagine di Skin utilizzata dal nostro algoritmo. È stata generata unendo pezzi di immagini di purè e polenta di vario tipo, raccolte sia dal dataset che da internet. Le illustrazioni seguenti invece mostrano una computazione dell'algoritmo sull'immagine "005.png".





Spazio Colore HSV

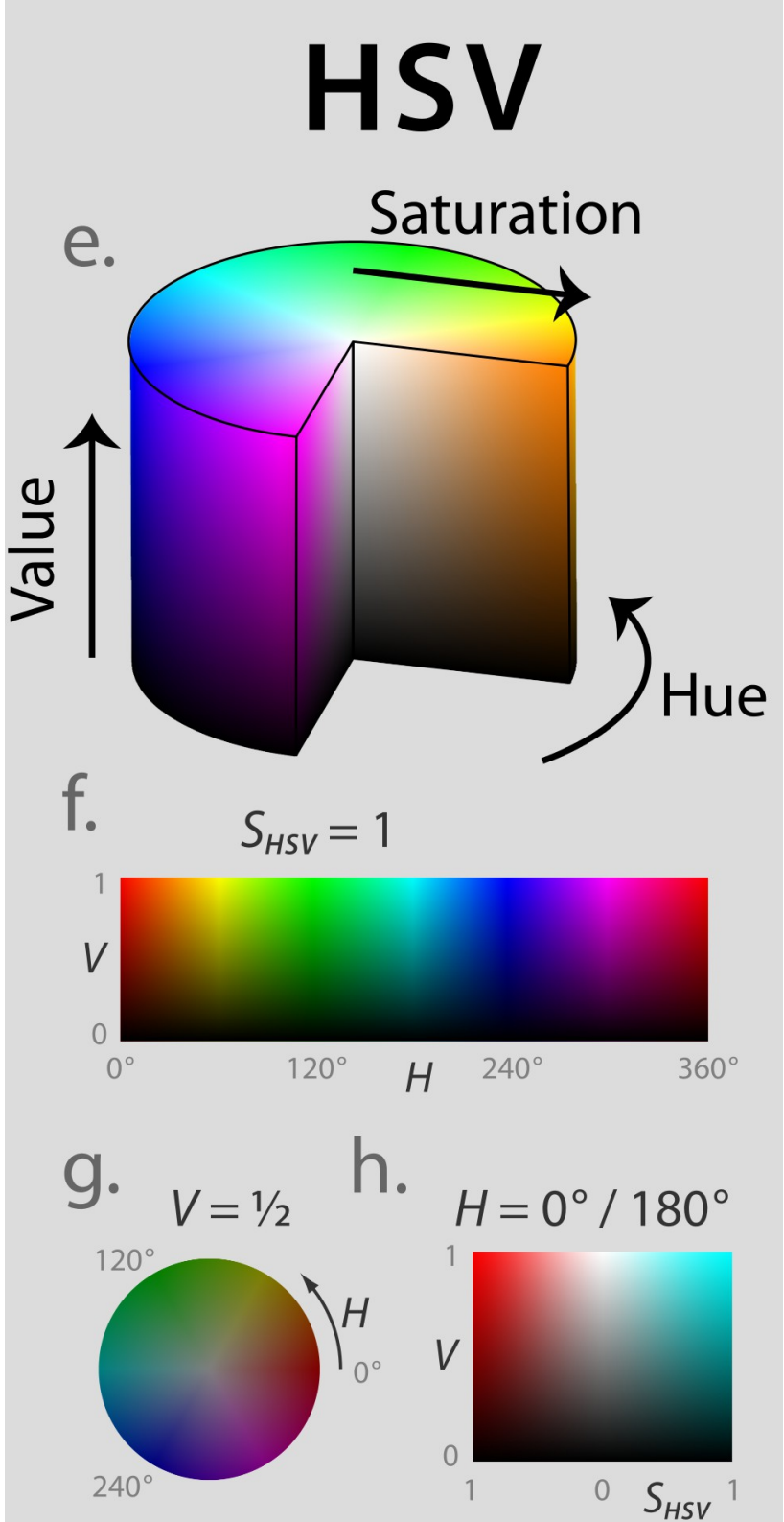
• Lo spazio colore HSV varia la disposizione dei colori in base a 3 valori caratteristici, di cui 2 lineari e 1 angolare;

• S indica la "Saturazione" del colore, ossia la sua percentuale di bianco;

• V è il valore che porta le informazioni relative alla luminosità del colore, ovvero la sua tendenza al bianco o al nero;

• H indica la tinta desiderata;

• Abbiamo scelto questo spazio colore perchè ci ha permesso di discriminare meglio gli oggetti d'interesse rispetto allo spazio colore RGB che ha una forte correlazione tra componenti. Scegliendo questo spazio abbiamo potuto "isolare" la componente intensità, sulla quale abbiamo lavorato.



Valori delle componenti HSV

La componente angolare H (in gradi)

0° oppure 360°	rosso
60°	giallo
120°	verde
180°	ciano
240°	blu
300°	magenta

La componente lineare V (con S=0)

1	bianco
(0...1)	grigi
0	nero

La componente lineare S (con V=1 e H=0)

1	colore saturo (puro)
(0...1)	saturazione intermedia
0	colore desaturato (grigio)

Riconoscimento Mandarini

- Binarizzazione con soglia 0.65;
- Differenza tra due bwareaopen rispettivamente con 5000 e 9000 pixel, per ricavare le sole regioni d'interesse;
- Discriminazione per area;
- Determinazione delle componenti connesse;
- Display della bounding box.

Immagine binarizzata e modificata con morfologia

output finale

Dettagli procedura riconoscimento mandarini

- Per binarizzare le immagini usiamo la funzione matlab im2bw, utilizzando come parametro la soglia 0.65. La soglia è stata scelta osservando che i contenitori dei mandarini sono bianchi.
- Due pixel sono connessi se, oltre ad essere adiacenti dal punto di vista spaziale, i loro livelli di grigio soddisfano un particolare criterio di similarità.
- Due tipi di connettività: la 4connettività e la 8connettività:

  – $N4(x,y)=\{(x-1,y),(x+1,y),(x,y-1),(x,y+1)\}$
  – $N8(x,y)=N4(x,y) \cup \{(x-1,y-1),(x-1,y+1),(x+1,y-1),(x+1,y+1)\}$

- Si parla di componente connessa se sono in possesso di una classe d'equivalenza di pixel che soddisfano il criterio di connettività.

Labeling delle Componenti Connesse

• Prima Scansione: Vengono assegnate delle label temporanee a tutti i pixel etichettati come oggetti in funzione delle label dei vicini già visitati. Oggetti distinti sono di certo stati separati ed etichettati con label diverse, ma anche parti di uno stesso oggetto potrebbero aver ricevuto label diverse;

• Seconda Scansione: Assegna una label definitiva univoca alle parti di uno stesso oggetto aventi label temporanee diverse. Fra le due scansioni è necessario individuare le label temporanee equivalenti ed assegnare a ciascuna classe di equivalenza una label definitiva univoca.

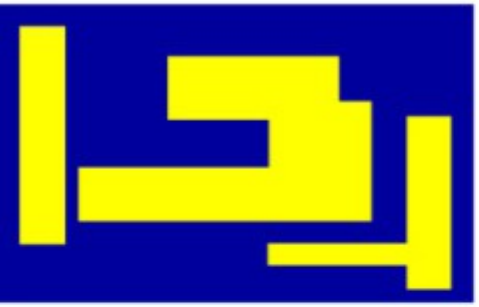
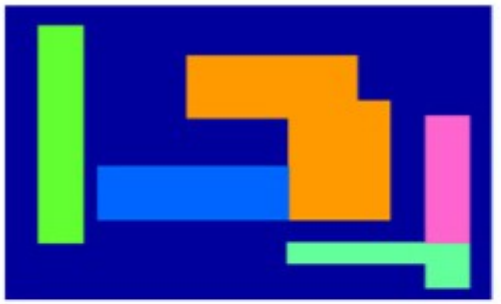
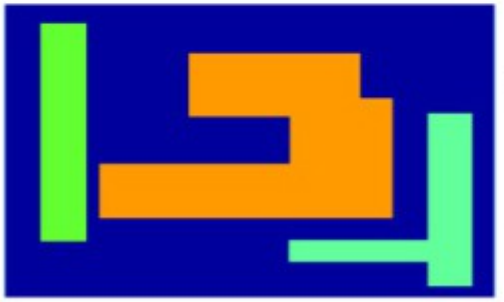


Immagine Originale

Esito della prima scansione



Definizione di una tabella di equivalenze



Esito della seconda scansione

Costruzione dell'algoritmo

- Per determinare il contenitore dei mandarini abbiamo utilizzato il rapporto fra asse maggiore e asse minore della regione e la solidity (restituisce uno scalare e viene ottenuta dal rapporto dell'area e dell'area convessa). Proprietà ricavate con la funzione matlab regionprops(BW, properties). La discriminazione è avvenuta provando vari valori ed osservando il comportamento dell'algoritmo.

```
BW2 = ismember(labeled, find([s.MajorAxisLength]./
[s.MinorAxisLength] >= 1.38 & [s.Solidity] >= 0.83));
```

Risultati Complessivi

- Per misurare l'efficienza dell'algoritmo i test sono stati effettuati su un campione del training set fornito.
- Abbiamo calcolato il totale di Bicchieri, Piatti, Mandarini e Purè presenti nelle prime 50 immagini del training set e lanciato in seguito la computazione. Per ogni immagine analizzata abbiamo calcolato gli elementi correttamente riconosciuti, i falsi positivi ed i falsi negativi. Nel lucido seguente si possono osservare i dati raccolti.

Risultati Complessivi

	Bicchieri	**Purè / Polenta**	**Mandarini**	**Piatti circolari o scodelle**
Oggetti Presenti	68	13	18	106
Oggetti Riconosciuti	68	11	16	105
Falsi Negativi	0	1	3	1
Falsi Positivi	1	6	0	0
Precisione	98,5%	64,7%	100%	100%

Analisi del nostro algoritmo

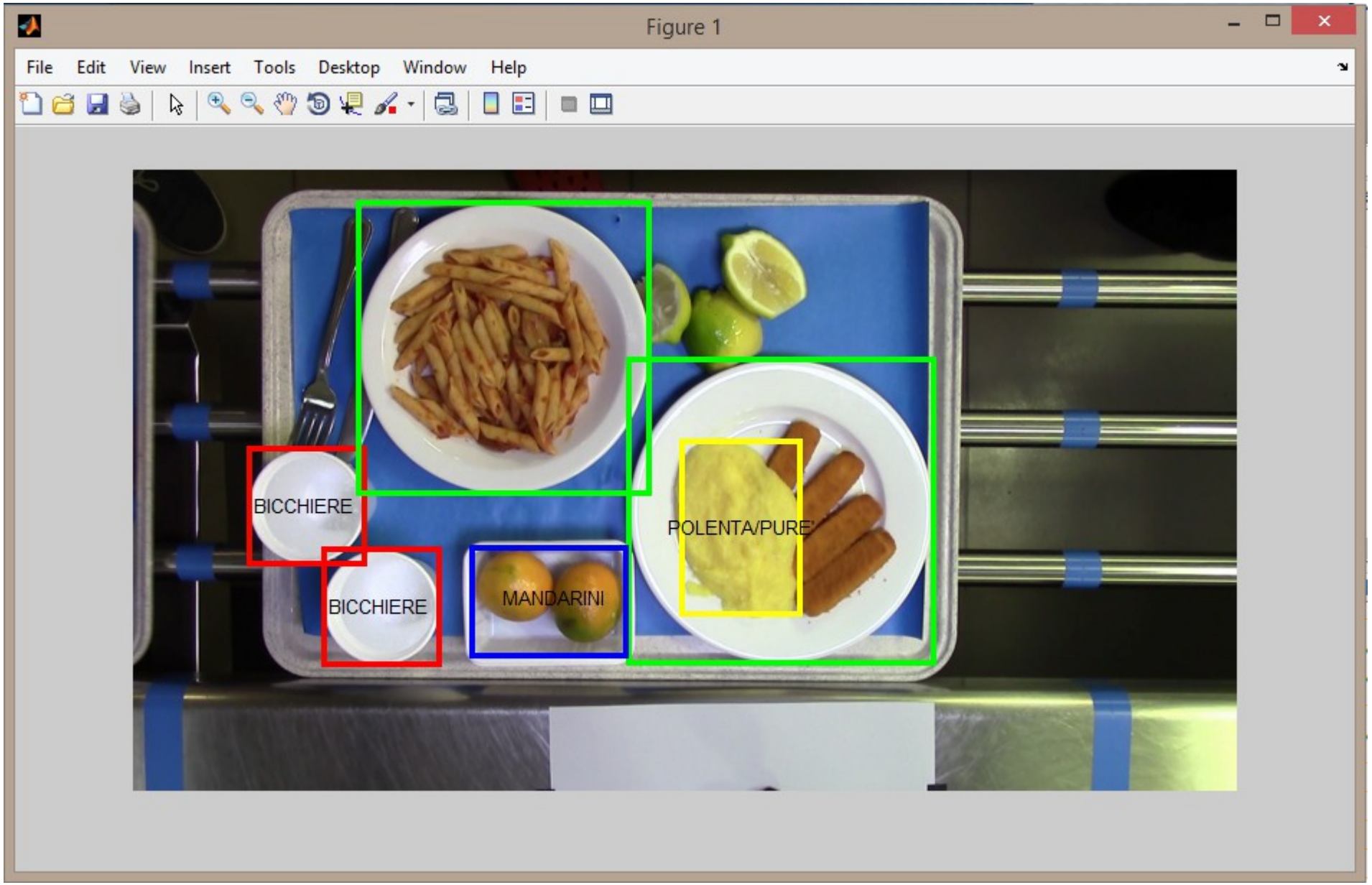
Pro

- Ottima velocità d'esecuzione, la durata media della computazione per ogni immagine è di circa 1,716 secondi (con alimentazione da rete elettrica) e di 3,285 secondi (con alimentazione a batteria);
- Feature stabili e discriminanti ai fini di un riconoscimento rapido e preciso;
- Vengono riconosciute più occorrenze di piatti e bicchieri all'interno della stessa immagine anche se le forme sono parzialmente occluse da altri oggetti presenti sul vassoio (Es. 031.png o 041.png);
- La presenza di oggetti estranei sul vassoio (Es. Telefonini) non creano problemi e non mandano in crash il programma;
- Riconoscimento di piatti circolari di varie dimensioni, feature indipendenti dal materiale e dal colore del piatto (Es. Nessuna distinzione tra piatti di plastica o di ceramica);
- Codice abbastanza leggibile ed interpretabile grazie ad alcune funzioni predefinite in Matlab (utile per un eventuale Refactoring);

Analisi del nostro algoritmo
Contro

- I Bicchieri non vengono riconosciuti nel caso in cui siano caduti sul vassoio ed alcune volte si presentano dei falsi negativi a causa della presenza nell'immagine di vasetti di yogurt (raggio e circonferenza simile a quella dei bicchieri);
- Nei test effettuati è stato riscontrato un numero non allarmante ma pur sempre presente di falsi positivi per il purè. Quasi sempre i falsi positivi riconosciuti sono in realtà i panini presenti sui vassoi o le pennette in bianco. Il problema potrebbe essere risolvibile ampliando l'immagine di skin della polenta o utilizzando altre feature o algoritmi di classificazione;
- Nei casi in cui la scatola di mandarini si trova in posizione obliqua a volte non viene riconosciuta correttamente e non viene tracciata dal programma;

Risultato Finale dell'algoritmo



Cos'altro avrei potuto riconoscere???

- All'interno delle immagini avrei potuto riconoscere ulteriori elementi, come ad esempio:

  - Carote;

  - Prosciutto cotto;

- Per farlo avrei potuto usare le medesime feature ed i medesimi algoritmi per entrambi gli elementi. Sarebbe bastato usare un classificatore non paramentrico come il KNN e sfruttare la feature "colore" dei 3 oggetti. Tuttavia è probabile che per le carote restituisse una serie molto alta di falsi negativi poichè il colore è simile a quello della pasta e dei bastoncini Findus ad esempio. Per il riconoscimento avrei potuto utilizzare ulteriori immagini di skin e no-skin per ogni oggetto.

Classificatore K Nearest Neighbour (Knn)

- La regola Knn determina i K elementi più vicini al pattern X da classificare. Ogni pattern tra i K vicini "vota" per la classe cui esso stesso appartiene ed infine X viene assegnato alla classe che ha ottenuto il maggior numero di voti.
- Pro: semplice, accurato, non ha bisogno di addestramento;
- Contro: costo computazionale legato alla fase di classificazione (può essere elevato).

Bibliografia

- Le informazioni e le immagini di queste slide sono state prese dalle seguenti fonti:
- http://it.mathworks.com/help/
- http://klee.cittastudi.di.unimi.it/~dan/grafica/doc/esame/argomenti/tursi-spazicolori/Sito/hsv.htm
- Slide delle lezioni
- Quaderno d'appunti presi in aula
- Dispense varie trovate su internet (situate nella cartella del progetto)

Fine della Presentazione